# Orthonormal basis

A basis $(u_i)_{i=1}^{n}$ is said to be /orthogonal/ if $u_i^T u_j = 0$ if $i \neq j$. If in addition, $\|u_i\|_2 = 1$, we say that the basis is *orthonormal*.

*Example:* An orthonormal basis in $\mathbf{R}^3$. The collection of vectors $\{u_1, u_2\}$, with

$$u_1 = \frac{1}{\sqrt{2}} \left( \begin{array}{c} 1 \\ 1 \end{array} \right), \quad u_2 = \frac{1}{\sqrt{2}} \left( \begin{array}{c} 1 \\ -1 \end{array} \right),$$

forms an orthonormal basis of $\mathbf{R}^2$.

# What is orthogonalization?

*Orthogonalization* refers to a procedure that finds an orthonormal basis of the span of given vectors.

Given vectors $a_1, \ldots, a_k \in \mathbf{R}^n$, an orthogonalization procedure computes vectors $q_1, \ldots, q_n \in \mathbf{R}^n$ such that

$$S := \operatorname{span}\{a_1, \ldots, a_k\} = \operatorname{span}\{q_1, \ldots, q_r\},$$

where $r$ is the dimension of $S$, and

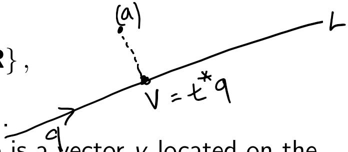
$$q_i^T q_j = 1 \ \ (i \neq j), \ \ q_i^T q_i = 1, \ \ 1 \leq i, j \leq r.$$

That is, the vectors $(q_1, \ldots, q_r)$ form an orthonormal basis for the span of the vectors $a_1, \ldots, a_k$.

# Projection on a line

A basic step in the procedure consists in projecting a vector on a line passing through zero. Consider the line

$$L(q) := \{tq \;:\; t \in \mathbf{R}\},$$




where $q \in \mathbf{R}^n$ is given, and normalized ($\|q\|_2 = 1$).
The *projection* of a given point $a \in \mathbf{R}^n$ on the line is a vector $v$ located on the line, that is closest to $a$ (in Euclidean norm). This corresponds to a simple optimization problem:

$$\min_t \; \|a - tq\|_2.$$

The vector $a_{\mathrm{proj}} := t^* q$, where $t^*$ is the optimal value, is referred to as the /projection/ of $a$ on the line $L(q)$. The solution of this simple problem has a closed-form expression:

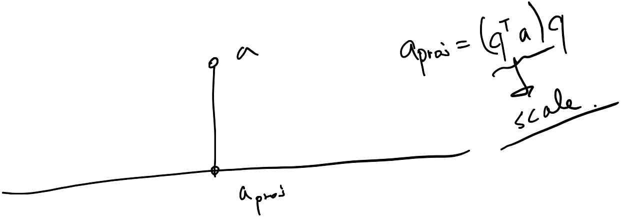
$$a_{\mathrm{proj}} = (q^T a) q.$$

# Interpretation

Note that the vector $x$ can now be written as a sum of its projection and another vector that is orthogonal to the projection:

$$a = (a - a_{\mathrm{proj}}) + a_{\mathrm{proj}} = (a - (q^T a)q) + (q^T a)q,$$

where $a - a_{\mathrm{proj}} = a - (q^T a)q$ and $a_{\mathrm{proj}} = (q^T a)q$ are orthogonal. The vector $a - a_{\mathrm{proj}}$ can be interpreted as the result of *removing the component* of $a$ along $q$.



# Gram-Schmidt procedure

The Gram-Schmidt procedure is a particular orthogonalization algorithm. The basic idea is to first orthogonalize each vector w.r.t. previous ones; then normalize result to have norm one.

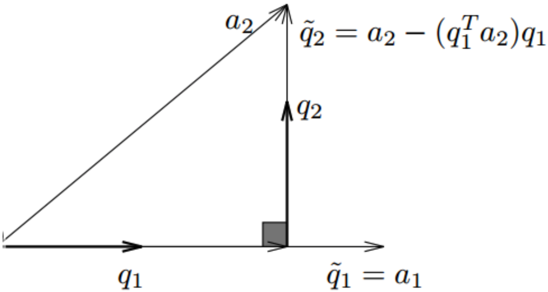
Let us assume that the vectors $a_1, \ldots, a_n$ are linearly independent. The GS algorithm is as follows.

**Gram-Schmidt procedure:**

1. set $\tilde{q}_1 = a_1$.
2. normalize: set $q_1 = \tilde{q}_1 / \|\tilde{q}_1\|_2$.
3. remove component of $q_1$ in $a_2$: set $\tilde{q}_2 = a_2 - (a_2^T q_1) q_1$.
4. normalize: set $q_2 = \tilde{q}_2 / \|\tilde{q}_2\|_2$.
5. remove components of $q_1, q_2$ in $a_3$: set $\tilde{q}_3 = a_3 - (a_3^T q_1) q_1 - (a_3^T q_2) q_2$.
6. normalize: set $q_3 = \tilde{q}_3 / \|\tilde{q}_3\|_2$.
7. etc.

The GS process is well-defined, since at each step $\tilde{q}_i \neq 0$ (otherwise this would contradict the linear independence of the $a_i$'s).

# GS in 2D




The image shows the GS procedure applied to the case of two vectors in two dimensions. We first set the first vector to be a normalized version of the first vector $a_1$. Then we remove the component of $a_2$ along the direction $q_1$. The difference is the (un-normalized) direction $\tilde{q}_2$, which becomes $q_2$ after normalization. At the end of the process, the vectors $q_1, q_2$ have both unit length and are orthogonal to each other.

# Geometry

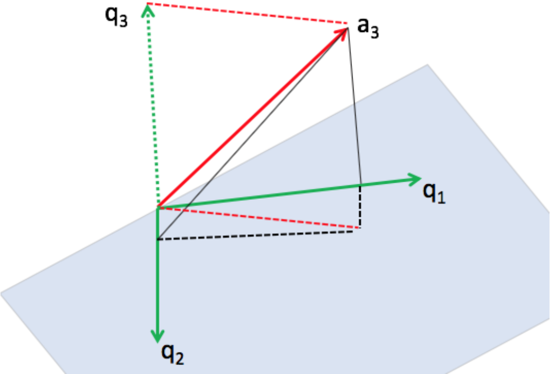



Figure: Geometry of QR: the third step in $\mathbb{R}^3$.

# Case with dependent vectors

It is possible to modify the algorithm to allow it to handle the case when the $a_i$'s are not linearly independent. If at step $i$, we find $\tilde{q}_i = 0$, then we directly jump at the next step.

**Modified Gram-Schmidt procedure:** set $r = 0$. for $i = 1, \ldots, n$:

1. set $\tilde{q} = a_i - \sum_{j=1}^{r}(q_j^T a_i)q_j$.
2. if $\tilde{q} \neq 0$, $r = r + 1$; $q_r = \tilde{q}/\|\tilde{q}\|_2$.

On exit, the integer $r$ is the dimension of the span of the vectors $a_1, \ldots, a_k$.

# QR decomposition

## Basic idea

The basic goal of the QR decomposition is to factor a matrix as a product of two matrices (traditionally called $Q, R$, hence the name of this factorization). Each matrix has a simple structure which can be further exploited in dealing with, say, linear equations.

The QR decomposition is nothing else than the Gram-Schmidt procedure applied to the columns of the matrix, and with the result expressed in matrix form.

# Full column rank case

Consider a $m \times n$ matrix $A = (a_1, \ldots, a_n)$, with each $a_i \in \mathbf{R}^m$ a column of $A$.

Assume first that the $a_i$'s (the columns of $A$) are linearly independent. That is, $A$ is full column-rank (its nullspace is $\{0\}$). Each step of the G-S procedure can be written as

$$a_i = (a_i^T q_1) q_1 + \ldots + (a_i^T q_{i-1}) q_{i-1} + \|\tilde{q}_i\|_2 q_i, \quad i = 1, \ldots, n.$$

We write this as

$$a_i = r_{i1} q_1 + \ldots + r_{i,i-1} q_{i-1} + r_{ii} q_i, \quad i = 1, \ldots, n,$$

where $r_{ij} = (a_i^T q_j)$ $(1 \leq j \leq i-1)$ and $r_{ii} = \|\tilde{q}_{ii}\|_2$.

# Full column rank case (cont'd)

Since the $q_i$'s are unit-length and normalized, the matrix $Q = (q_1, \ldots, q_n)$ satisfies $Q^T Q = I_n$. The QR decomposition of a $m \times n$ matrix $A$ thus allows to write the matrix in /factored/ form:

$$A = QR, \quad Q = \begin{pmatrix} q_1 & \ldots & q_n \end{pmatrix}, \quad R = \begin{pmatrix} r_{11} & r_{12} & \ldots & r_{1n} \\ 0 & r_{22} & & r_{2n} \\ \vdots & & \ddots & \vdots \\ 0 & & 0 & r_{nn} \end{pmatrix}$$

where $Q$ is a $m \times n$ matrix with $Q^T Q = I_n$, and $R$ is $n \times n$,upper-triangular.

# Example

$$A = \begin{pmatrix} 1 & 2 & 7 \\ 3 & 4 & 8 \\ 5 & 6 & 1 \end{pmatrix} = QR, \quad Q = \begin{pmatrix} -0.1690 & 0.8971 & 0.4082 \\ -0.5071 & 0.2760 & -0.8165 \\ -0.8452 & -0.3450 & 0.4082 \end{pmatrix},$$

$$R = \begin{pmatrix} -5.9161 & -7.4374 & -6.0851 \\ 0 & 0.8281 & 8.1428 \\ 0 & 0 & -3.2660 \end{pmatrix}.$$

# Case when the columns are not independent

When the columns of $A$ are not independent, at some step of the G-S procedure we encounter a zero vector $\tilde{q}_j$, which means $a_j$ is a linear combination of $a_{j-1}, \ldots, a_1$. The "modified" Gram-Schmidt procedure then simply skips to the next vector and continues.

In matrix form, we obtain $A = QR$, with $Q \in \mathbf{R}^{m \times r}$, $r = \mathbf{Rank}(A)$, and $R$ has an upper staircase form, for example:

$$R = \begin{pmatrix} * & * & * & * & * & * \\ 0 & 0 & * & * & * & * \\ 0 & 0 & 0 & 0 & * & * \end{pmatrix}.$$

(This is simply an upper triangular matrix with some rows deleted. It is still upper triangular.)

# Reordering

We can permute the columns of $R$ to bring forward the first non-zero elements in each row:

$$R = \begin{pmatrix} R_1 & R_2 \end{pmatrix} P^T, \quad \left( \begin{array}{c|c} R_1 & R_2 \end{array} \right) := \left( \begin{array}{ccc|ccc} * & * & * & * & * & * \\ 0 & * & 0 & * & * & * \\ 0 & 0 & * & 0 & 0 & * \end{array} \right),$$

where $P$ is a permutation matrix (that is, its columns are the unit vectors in some order), whose effect is to permute columns. (Since $P$ is orthogonal, $P^{-1} = P^T$.) Now, $R_1$ is square, upper triangular, and invertible, since none of its diagonal elements is zero.

# Reordering: matrix format

The QR decomposition can be written

$$AP = Q \begin{pmatrix} R_1 & R_2 \end{pmatrix},$$

where

1. 
2. $Q \in \mathbf{R}^{m\times r}$, $Q^T Q = I_r$;
3. $r$ is the rank of $A$;
4. $R_1$ is $r \times r$ upper triangular, invertible matrix;
5. $R_2$ is a $r \times (n-r)$ matrix;
6. $P$ is a $m \times m$ permutation matrix.