



University
of Manitoba

QR factorisation & SVD

MATH 2740 – Mathematics of Data Science – Unit U04

Julien Arino

`julien.arino@umanitoba.ca`

Department of Mathematics @ University of Manitoba

Fall 2026

The University of Manitoba campuses are located on original lands of Anishinaabeg, Ininew, Anisininew, Dakota and Dene peoples, and on the National Homeland of the Red River Métis. We respect the Treaties that were made on these territories, we acknowledge the harms and mistakes of the past, and we dedicate ourselves to move forward in partnership with Indigenous communities in a spirit of Reconciliation and collaboration.

Outline

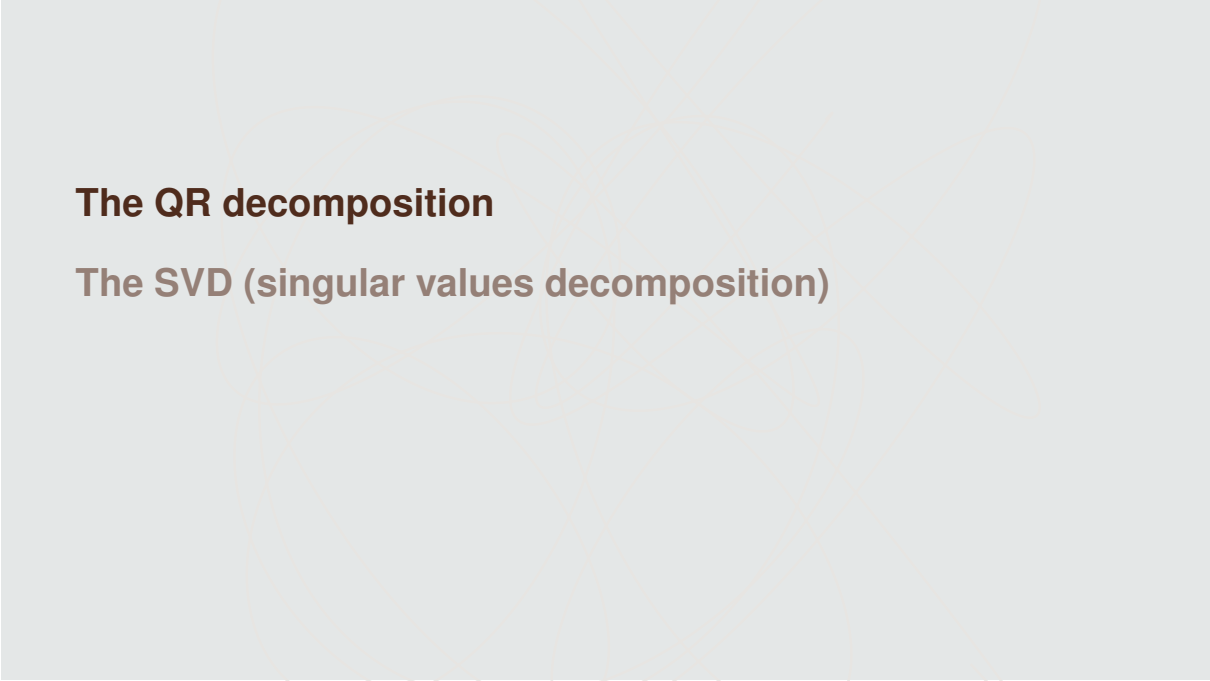
The QR decomposition

The SVD (singular values decomposition)

Matrix factorisations

Matrix factorisations are popular because they allow to perform some computations more easily

There are several different types of factorisations. Here, we study just the QR factorisation (useful for many least squares problems) and SVD (useful in many contexts)



The QR decomposition

The SVD (singular values decomposition)

The QR decomposition

Orthogonality and projections

The Gram-Schmidt orthogonalisation process

Orthogonal matrices

The QR factorisation & Least squares

Definition 57 (Orthogonal set of vectors)

The set of vectors $\{\mathbf{v}_1, \dots, \mathbf{v}_k\} \in \mathbb{R}^n$ is an **orthogonal set** if

$$\forall i, j = 1, \dots, k, \quad i \neq j \implies \mathbf{v}_i \bullet \mathbf{v}_j = 0$$

Theorem 58

$\{\mathbf{v}_1, \dots, \mathbf{v}_k\} \in \mathbb{R}^n$ with $\forall i, \mathbf{v}_i \neq \mathbf{0}$, orthogonal set $\implies \{\mathbf{v}_1, \dots, \mathbf{v}_k\} \in \mathbb{R}^n$ linearly independent

Definition 59 (Orthogonal basis)

Let S be a basis of the subspace $W \subset \mathbb{R}^n$ composed of an orthogonal set of vectors. We say S is an **orthogonal basis** of W

Proof of Theorem 58

Assume $\{\mathbf{v}_1, \dots, \mathbf{v}_k\}$ orthogonal set with $\mathbf{v}_i \neq \mathbf{0}$ for all $i = 1, \dots, k$. Recall $\{\mathbf{v}_1, \dots, \mathbf{v}_k\}$ is LI if

$$c_1 \mathbf{v}_1 + \dots + c_k \mathbf{v}_k = \mathbf{0} \iff c_1 = \dots = c_k = 0$$

So assume $c_1, \dots, c_k \in \mathbb{R}$ are s.t. $c_1 \mathbf{v}_1 + \dots + c_k \mathbf{v}_k = \mathbf{0}$. Recall that $\forall \mathbf{x} \in \mathbb{R}^k$, $\mathbf{0}_k \bullet \mathbf{x} = 0$. So for some $\mathbf{v}_i \in \{\mathbf{v}_1, \dots, \mathbf{v}_k\}$

$$\begin{aligned} 0 &= \mathbf{0} \bullet \mathbf{v}_i \\ &= (c_1 \mathbf{v}_1 + \dots + c_k \mathbf{v}_k) \bullet \mathbf{v}_i \\ &= c_1 \mathbf{v}_1 \bullet \mathbf{v}_i + \dots + c_k \mathbf{v}_k \bullet \mathbf{v}_i \end{aligned} \tag{1}$$

As $\{\mathbf{v}_1, \dots, \mathbf{v}_k\}$ orthogonal, $\mathbf{v}_j \bullet \mathbf{v}_i = 0$ when $i \neq j$, (1) reduces to

$$c_i \mathbf{v}_i \bullet \mathbf{v}_i = 0 \iff c_i \|\mathbf{v}_i\|^2 = 0$$

As $\mathbf{v}_i \neq \mathbf{0}$ for all i , $\|\mathbf{v}_i\| \neq 0$ and so $c_i = 0$. This is true for all i , hence the result □

Example – Vectors of the standard basis of $\mathbb{R}^3$

For $\mathbb{R}^3$, we denote

$$\mathbf{i} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \quad \mathbf{j} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \quad \text{and} \quad \mathbf{k} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

($\mathbb{R}^k$ for $k > 3$, we denote them $\mathbf{e}_i$)

Clearly, $\{\mathbf{i}, \mathbf{j}\}$, $\{\mathbf{i}, \mathbf{k}\}$, $\{\mathbf{j}, \mathbf{k}\}$ and $\{\mathbf{i}, \mathbf{j}, \mathbf{k}\}$ orthogonal sets. The standard basis vectors are also $\neq \mathbf{0}$, so the sets are LI. And

$$\{\mathbf{i}, \mathbf{j}, \mathbf{k}\}$$

is an orthogonal basis of $\mathbb{R}^3$ since it spans $\mathbb{R}^3$ and is LI

$$c_1 \mathbf{i} + c_2 \mathbf{j} + c_3 \mathbf{k} = c_1 \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} + c_2 \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} + c_3 \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} c_1 \\ c_2 \\ c_3 \end{pmatrix}$$

Orthonormal version of things

Definition 60 (Orthonormal set)

The set of vectors $\{\mathbf{v}_1, \dots, \mathbf{v}_k\} \in \mathbb{R}^n$ is an **orthonormal set** if it is an orthogonal set and furthermore

$$\forall i = 1, \dots, k, \quad \|\mathbf{v}_i\| = 1$$

Definition 61 (Orthonormal basis)

A basis of the subspace $W \subset \mathbb{R}^n$ is an **orthonormal basis** if the vectors composing it are an orthonormal set

$\{\mathbf{v}_1, \dots, \mathbf{v}_k\} \in \mathbb{R}^n$ is orthonormal if

$$\mathbf{v}_i \bullet \mathbf{v}_j = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}$$

The QR decomposition

Orthogonality and projections

The Gram-Schmidt orthogonalisation process

Orthogonal matrices

The QR factorisation & Least squares

What this aims to do is to construct an orthogonal basis for a subspace $W \subset \mathbb{R}^n$

To do this, we use the *Gram-Schmidt orthogonalisation process*, which turns a basis of W into an orthogonal basis of W

Projections

Definition 62 (Orthogonal projection onto a subspace)

$W \subset \mathbb{R}^n$ a subspace and $\{\mathbf{u}_1, \dots, \mathbf{u}_k\}$ an orthogonal basis of W . $\forall \mathbf{v} \in \mathbb{R}^n$, the **orthogonal projection** of $\mathbf{v}$ onto W is

$$\text{proj}_W(\mathbf{v}) = \frac{\mathbf{u}_1 \bullet \mathbf{v}}{\|\mathbf{u}_1\|^2} \mathbf{u}_1 + \dots + \frac{\mathbf{u}_k \bullet \mathbf{v}}{\|\mathbf{u}_k\|^2} \mathbf{u}_k$$

Definition 63 (Component orthogonal to a subspace)

$W \subset \mathbb{R}^n$ a subspace and $\{\mathbf{u}_1, \dots, \mathbf{u}_k\}$ an orthogonal basis of W . $\forall \mathbf{v} \in \mathbb{R}^n$, the **component** of $\mathbf{v}$ orthogonal to W is

$$\text{perp}_W(\mathbf{v}) = \mathbf{v} - \text{proj}_W(\mathbf{v})$$

Theorem 64 (Gram-Schmidt orthogonalisation process)

Let W be a subspace of $\mathbb{R}^n$ and $\{\mathbf{x}_1, \dots, \mathbf{x}_k\}$ be a basis of W . Let

$$\mathbf{v}_1 = \mathbf{x}_1$$

$$\mathbf{v}_2 = \mathbf{x}_2 - \frac{\mathbf{v}_1 \bullet \mathbf{x}_2}{\|\mathbf{v}_1\|^2} \mathbf{v}_1$$

$$\mathbf{v}_3 = \mathbf{x}_3 - \frac{\mathbf{v}_1 \bullet \mathbf{x}_3}{\|\mathbf{v}_1\|^2} \mathbf{v}_1 - \frac{\mathbf{v}_2 \bullet \mathbf{x}_3}{\|\mathbf{v}_2\|^2} \mathbf{v}_2$$

$\vdots$

$$\mathbf{v}_k = \mathbf{x}_k - \frac{\mathbf{v}_1 \bullet \mathbf{x}_k}{\|\mathbf{v}_1\|^2} \mathbf{v}_1 - \dots - \frac{\mathbf{v}_{k-1} \bullet \mathbf{x}_k}{\|\mathbf{v}_{k-1}\|^2} \mathbf{v}_{k-1}$$

and

$$W_1 = \text{span}(\mathbf{x}_1), W_2 = \text{span}(\mathbf{x}_1, \mathbf{x}_2), \dots, W_k = \text{span}(\mathbf{x}_1, \dots, \mathbf{x}_k)$$

Then $\forall i = 1, \dots, k$, $\{\mathbf{v}_1, \dots, \mathbf{v}_i\}$ is an orthogonal basis of W_i

Theorem 65 (Gram-Schmidt orthonormalisation process)

Let W be a subspace of $\mathbb{R}^n$ and $\{\mathbf{x}_1, \dots, \mathbf{x}_k\}$ be a basis of W . Let

$$\tilde{\mathbf{v}}_1 = \mathbf{x}_1 / \|\mathbf{x}_1\|$$

$$\mathbf{v}_2 = \mathbf{x}_2 - (\tilde{\mathbf{v}}_1 \bullet \mathbf{x}_2) \tilde{\mathbf{v}}_1$$

$$\tilde{\mathbf{v}}_2 = \frac{\mathbf{v}_2}{\|\mathbf{v}_2\|}$$

$$\mathbf{v}_3 = \mathbf{x}_3 - (\tilde{\mathbf{v}}_1 \bullet \mathbf{x}_3) \tilde{\mathbf{v}}_1 - (\tilde{\mathbf{v}}_2 \bullet \mathbf{x}_3) \tilde{\mathbf{v}}_2$$

$$\tilde{\mathbf{v}}_3 = \frac{\mathbf{v}_3}{\|\mathbf{v}_3\|}$$

$\vdots$

$$\mathbf{v}_k = \mathbf{x}_k - (\tilde{\mathbf{v}}_1 \bullet \mathbf{x}_k) \tilde{\mathbf{v}}_1 - \dots - (\tilde{\mathbf{v}}_{k-1} \bullet \mathbf{x}_k) \tilde{\mathbf{v}}_{k-1}$$

$$\tilde{\mathbf{v}}_k = \frac{\mathbf{v}_k}{\|\mathbf{v}_k\|}$$

and

$$W_1 = \text{span}(\mathbf{x}_1), W_2 = \text{span}(\mathbf{x}_1, \mathbf{x}_2), \dots, W_k = \text{span}(\mathbf{x}_1, \dots, \mathbf{x}_k)$$

Then $\forall i = 1, \dots, k$, $\{\tilde{\mathbf{v}}_1, \dots, \tilde{\mathbf{v}}_i\}$ is an orthonormal basis of W_i

Links between the two processes

- ▶ To get an orthonormal basis, you can also use the orthogonalisation process then normalise the result
- ▶ There is no difference between the two methods, the only gain in the orthonormalisation process is that you don't need to divide by $\|\cdot\|^2$ when projecting, since you have already done so just before

The QR decomposition

Orthogonality and projections

The Gram-Schmidt orthogonalisation process

Orthogonal matrices

The QR factorisation & Least squares

Theorem 66

Let $Q \in \mathcal{M}_{np}$. The columns of Q form an orthonormal set if and only if

$$Q^T Q = \mathbb{I}_p$$

Definition 67 (Orthogonal matrix)

$Q \in \mathcal{M}_n$ is an **orthogonal matrix** if its columns form an orthonormal set

So $Q \in \mathcal{M}_n$ orthogonal if $Q^T Q = \mathbb{I}$, i.e., $Q^T = Q^{-1}$

Theorem 68 (NSC for orthogonality)

$Q \in \mathcal{M}_n$ orthogonal $\iff Q^{-1} = Q^T$

Orthogonal matrices “encode” isometries

Theorem 69 (Orthogonal matrices “encode” isometries)

Let $Q \in \mathcal{M}_n$. TFAE

1. Q orthogonal
2. $\forall \mathbf{x} \in \mathbb{R}^n, \|Q\mathbf{x}\| = \|\mathbf{x}\|$
3. $\forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^n, Q\mathbf{x} \bullet Q\mathbf{y} = \mathbf{x} \bullet \mathbf{y}$

Proof of Theorem 69

1 $\implies$ 3: Assume Q is orthogonal. Then $Q^T Q = \mathbb{I}_n$, so

$$Q\mathbf{x} \bullet Q\mathbf{y} = (Q\mathbf{x})^T (Q\mathbf{y}) = \mathbf{x}^T Q^T Q \mathbf{y} = \mathbf{x}^T \mathbb{I}_n \mathbf{y} = \mathbf{x} \bullet \mathbf{y}$$

3 $\implies$ 1: Assume $Q\mathbf{x} \bullet Q\mathbf{y} = \mathbf{x} \bullet \mathbf{y} \forall \mathbf{x}, \mathbf{y}$. For standard basis vectors $\mathbf{e}_i, \mathbf{e}_j$:

$$(Q\mathbf{e}_i) \bullet (Q\mathbf{e}_j) = \mathbf{e}_i \bullet \mathbf{e}_j = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}$$

Since $Q\mathbf{e}_i$ is the i -th column of Q , the columns of Q are orthonormal, so Q is orthogonal.

3 $\implies$ 2: Assume $Q\mathbf{x} \bullet Q\mathbf{y} = \mathbf{x} \bullet \mathbf{y}$. Letting $\mathbf{y} = \mathbf{x}$ gives

$$\|Q\mathbf{x}\|^2 = Q\mathbf{x} \bullet Q\mathbf{x} = \mathbf{x} \bullet \mathbf{x} = \|\mathbf{x}\|^2 \implies \|Q\mathbf{x}\| = \|\mathbf{x}\|$$

2 $\implies$ 3: Assume $\|Q\mathbf{z}\| = \|\mathbf{z}\|$ for all $\mathbf{z}$. Using $\mathbf{z} = \mathbf{x} + \mathbf{y}$, we expand $\|Q(\mathbf{x} + \mathbf{y})\|^2 = \|\mathbf{x} + \mathbf{y}\|^2$:

$$\|Q\mathbf{x}\|^2 + 2Q\mathbf{x} \bullet Q\mathbf{y} + \|Q\mathbf{y}\|^2 = \|\mathbf{x}\|^2 + 2\mathbf{x} \bullet \mathbf{y} + \|\mathbf{y}\|^2$$

Since $\|Q\mathbf{x}\| = \|\mathbf{x}\|$ and $\|Q\mathbf{y}\| = \|\mathbf{y}\|$, we get $Q\mathbf{x} \bullet Q\mathbf{y} = \mathbf{x} \bullet \mathbf{y}$.



Theorem 70

Let $Q \in \mathcal{M}_n$ be orthogonal. Then

- 1. The rows of Q form an orthonormal set*
- 2. Q^{-1} orthogonal*
- 3. $\det Q = \pm 1$*
- 4. $\forall \lambda \in \sigma(Q), |\lambda| = 1$*
- 5. If $Q_2 \in \mathcal{M}_n$ also orthogonal, then QQ_2 orthogonal*

Proof of 1 and 2 in Theorem 70

1. The rows of Q form an orthonormal set

- ▶ By definition, Q is orthogonal $\iff Q^T Q = \mathbb{I}_n$
- ▶ Since Q is square, $Q^T Q = \mathbb{I}_n \implies Q^{-1} = Q^T \implies QQ^T = \mathbb{I}_n$
- ▶ This is equivalent to $(Q^T)^T Q^T = \mathbb{I}_n$, so the columns of Q^T (the rows of Q) are orthonormal □

2. Q^{-1} is orthogonal

- ▶ We must show that $(Q^{-1})^T Q^{-1} = \mathbb{I}_n$
- ▶ Using $Q^{-1} = Q^T$, we get $(Q^{-1})^T Q^{-1} = (Q^T)^T Q^T = QQ^T = \mathbb{I}_n$
- ▶ Thus Q^{-1} is an orthogonal matrix □

Proof of 3 and 5 in Theorem 70

3. $\det Q = \pm 1$

- ▶ We have $Q^T Q = \mathbb{I}_n$. Taking determinant on both sides

$$\det(Q^T Q) = \det(\mathbb{I}_n) = 1$$

- ▶ Using $\det(AB) = \det(A)\det(B)$, we get $\det(Q^T)\det(Q) = 1$
- ▶ From the general property $\det(A^T) = \det(A)$, it follows that $(\det(Q))^2 = 1$, so $\det Q = \pm 1$ □

5. If $Q_2 \in \mathcal{M}_n$ also orthogonal, then QQ_2 orthogonal

- ▶ We must show that $(QQ_2)^T(QQ_2) = \mathbb{I}_n$
- ▶ $(QQ_2)^T(QQ_2) = Q_2^T Q^T QQ_2 = Q_2^T (Q^T Q) Q_2$
- ▶ Since Q is orthogonal, $Q^T Q = \mathbb{I}_n$, so the expression becomes $Q_2^T \mathbb{I}_n Q_2 = Q_2^T Q_2$
- ▶ Since Q_2 is orthogonal, $Q_2^T Q_2 = \mathbb{I}_n$ □

Proof of 4 in Theorem 70

4. $\forall \lambda \in \sigma(Q), |\lambda| = 1$

Let λ be an eigenvalue of $Q \in \mathcal{M}_n$ orthogonal, i.e., $\exists \mathbb{R}^n \ni \mathbf{x} \neq \mathbf{0}$ s.t.

$$Q\mathbf{x} = \lambda\mathbf{x}$$

Take the norm on both sides

$$\|Q\mathbf{x}\| = \|\lambda\mathbf{x}\|$$

From 2 in Theorem 69, $\|Q\mathbf{x}\| = \|\mathbf{x}\|$ and from the properties of norms, $\|\lambda\mathbf{x}\| = |\lambda| \|\mathbf{x}\|$, so we have

$$\|Q\mathbf{x}\| = \|\lambda\mathbf{x}\| \iff \|\mathbf{x}\| = |\lambda| \|\mathbf{x}\| \iff 1 = |\lambda|$$

(we can divide by $\|\mathbf{x}\|$ since $\mathbf{x} \neq \mathbf{0}$ as an eigenvector)



The QR decomposition

Orthogonality and projections

The Gram-Schmidt orthogonalisation process

Orthogonal matrices

The QR factorisation & Least squares

The QR factorisation

Theorem 71

Let $A \in \mathcal{M}_{np}$ with LI columns. Then A can be factored as

$$A = QR$$

where $Q \in \mathcal{M}_{np}$ has orthonormal columns and $R \in \mathcal{M}_p$ is nonsingular upper triangular

Back to least squares

So what was the point of all that..?

Theorem 72 (Least squares with QR factorisation)

$A \in \mathcal{M}_{np}$ with linearly independent columns, $\mathbf{b} \in \mathbb{R}^n$. If $A = QR$ is a QR factorisation of A , then the unique least squares solution $\tilde{\mathbf{x}}$ of $A\mathbf{x} = \mathbf{b}$ is

$$\tilde{\mathbf{x}} = R^{-1} Q^T \mathbf{b}$$

Proof of Theorem 72

A has LI columns so

- ▶ least squares $A\mathbf{x} = \mathbf{b}$ has unique solution $\tilde{\mathbf{x}} = (A^T A)^{-1} A^T \mathbf{b}$
- ▶ by Theorem 71, A can be written as $A = QR$ with $Q \in \mathcal{M}_{np}$ with orthonormal columns and $R \in \mathcal{M}_p$ nonsingular and upper triangular

So

$$\begin{aligned} A^T A \tilde{\mathbf{x}} &= A^T \mathbf{b} \implies (QR)^T QR \tilde{\mathbf{x}} = (QR)^T \mathbf{b} \\ &\implies R^T Q^T QR \tilde{\mathbf{x}} = R^T Q^T \mathbf{b} \\ &\implies R^T \mathbb{I}_p R \tilde{\mathbf{x}} = R^T Q^T \mathbf{b} \\ &\implies R^T R \tilde{\mathbf{x}} = R^T Q^T \mathbf{b} \\ &\implies (R^T)^{-1} R \tilde{\mathbf{x}} = (R^T)^{-1} R^T Q^T \mathbf{b} \\ &\implies R \tilde{\mathbf{x}} = Q^T \mathbf{b} \\ &\implies \tilde{\mathbf{x}} = R^{-1} Q^T \mathbf{b} \quad \square \end{aligned}$$

Other applications of the QR decomposition

Focus here on least squares, but QR decomposition fundamental tool in numerical linear algebra with other important applications

- ▶ **Solving linear systems:** to solve $A\mathbf{x} = \mathbf{b}$ for a non-singular square matrix A , write $QR\mathbf{x} = \mathbf{b} \implies R\mathbf{x} = Q^T\mathbf{b}$. Since R is upper triangular, this can be quickly solved by back-substitution. This method is often more numerically stable than Gaussian elimination (called the LU decomposition)
- ▶ **Computing eigenvalues:** the **QR algorithm** is a standard method for calculating the eigenvalues of a matrix. It uses the iterative process $A_k = Q_k R_k$ and $A_{k+1} = R_k Q_k$. Under certain conditions, A_k converges to an upper triangular matrix whose diagonal entries are the eigenvalues
- ▶ **Computing determinants:** if $A = QR$, then $\det(A) = \det(Q)\det(R)$. Since Q is orthogonal, $\det(Q) = \pm 1$, and since R is upper triangular, $\det(R)$ is simply the product of its diagonal entries

Example – Solving least squares with QR (1/5)

Consider the matrix A and a vector $\mathbf{b}$:

$$A = \begin{pmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} \quad \text{and} \quad \mathbf{b} = \begin{pmatrix} 2 \\ 3 \\ 4 \end{pmatrix}.$$

We want to find the unique least squares solution $\tilde{\mathbf{x}}$ to $A\mathbf{x} = \mathbf{b}$.

To do this using the QR decomposition, we must first compute Q and R .

Example – Solving least squares with QR (2/5)

We compute Q by applying the Gram-Schmidt process to the columns of A :

$\mathbf{a}_1 = (1, 1, 0)$ and $\mathbf{a}_2 = (1, 0, 1)$.

First, we find the orthogonal vectors:

$$\mathbf{v}_1 = \mathbf{a}_1 = (1, 1, 0).$$

$$\mathbf{v}_2 = \mathbf{a}_2 - \frac{\mathbf{v}_1 \bullet \mathbf{a}_2}{\|\mathbf{v}_1\|^2} \mathbf{v}_1 = (1, 0, 1) - \frac{1}{2}(1, 1, 0) = \left(\frac{1}{2}, -\frac{1}{2}, 1\right).$$

Then, we normalise them to obtain the orthonormal columns of Q :

$$\mathbf{u}_1 = \frac{\mathbf{v}_1}{\|\mathbf{v}_1\|} = \left(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}, 0\right).$$

$$\mathbf{u}_2 = \frac{\mathbf{v}_2}{\|\mathbf{v}_2\|} = \sqrt{\frac{2}{3}} \left(\frac{1}{2}, -\frac{1}{2}, 1\right) = \left(\frac{1}{\sqrt{6}}, -\frac{1}{\sqrt{6}}, \frac{2}{\sqrt{6}}\right).$$

Example – Solving least squares with QR (3/5)

Thus, the orthogonal matrix Q is:

$$Q = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{6}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{6}} \\ 0 & \frac{2}{\sqrt{6}} \end{pmatrix}.$$

To find R , recall that $A = QR$. Since Q has orthonormal columns, $Q^T Q = \mathbb{I}$. Thus:

$$Q^T A = Q^T QR = \mathbb{I}R = R.$$

We can therefore compute R directly as $Q^T A$:

$$R = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ \frac{1}{\sqrt{6}} & -\frac{1}{\sqrt{6}} & \frac{2}{\sqrt{6}} \end{pmatrix} \begin{pmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} \frac{2}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ 0 & \frac{3}{\sqrt{6}} \end{pmatrix}.$$

Example – Solving least squares with QR (4/5)

By Theorem 72, we know that $R\tilde{\mathbf{x}} = Q^T \mathbf{b}$. First, we compute $Q^T \mathbf{b}$:

$$Q^T \mathbf{b} = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ \frac{1}{\sqrt{6}} & -\frac{1}{\sqrt{6}} & \frac{2}{\sqrt{6}} \end{pmatrix} \begin{pmatrix} 2 \\ 3 \\ 4 \end{pmatrix} = \begin{pmatrix} \frac{5}{\sqrt{2}} \\ \frac{7}{\sqrt{6}} \end{pmatrix}.$$

Now we solve the upper triangular system $R\tilde{\mathbf{x}} = Q^T \mathbf{b}$ via back-substitution:

$$\begin{pmatrix} \frac{2}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ 0 & \frac{3}{\sqrt{6}} \end{pmatrix} \begin{pmatrix} \tilde{x}_1 \\ \tilde{x}_2 \end{pmatrix} = \begin{pmatrix} \frac{5}{\sqrt{2}} \\ \frac{7}{\sqrt{6}} \end{pmatrix}.$$

The second equation gives $\frac{3}{\sqrt{6}} \tilde{x}_2 = \frac{7}{\sqrt{6}}$, which implies $\tilde{x}_2 = \frac{7}{3}$.

Example – Solving least squares with QR (5/5)

Substituting this into the first equation yields $\frac{2}{\sqrt{2}}\tilde{x}_1 + \frac{1}{\sqrt{2}}\left(\frac{7}{3}\right) = \frac{5}{\sqrt{2}}$, which simplifies to $2\tilde{x}_1 + \frac{7}{3} = 5$. Solving for $\tilde{x}_1$ yields $\tilde{x}_1 = \frac{4}{3}$.

Thus, the least squares solution is $\tilde{\mathbf{x}} = \begin{pmatrix} 4/3 \\ 7/3 \end{pmatrix}$.

We can easily confirm this result in R:

```
b <- c(2, 3, 4)
QT_b <- t(Q_A) %*% b
x_tilde <- solve(R_A, QT_b)
x_tilde

##           [,1]
## [1,] 1.333333
## [2,] 2.333333

qr.coef(qr_A, b) # with the built-in solver

## [1] 1.333333 2.333333
```

Fitting the Canada census data using QR

Let us return to the Canada census data we fitted earlier using the normal equations.

We saw that computing $A^T A$ and its inverse for a quadratic fit was problematic when using the raw years, because the matrix A was badly conditioned.

This led us to shift the x -axis to start at $t = 0$.

The QR decomposition provides a much more numerically stable way to solve the least squares problem $A\mathbf{x} = \mathbf{b}$.

To illustrate this, let us try to fit the quadratic model on the unshifted raw data using the QR approach.

Setup for unshifted quadratic fit

We first load the data and set up our matrix A for a quadratic fit without shifting the years.

```
data <- read.csv("../DATA/Canada_census.csv")
data <- data[which(data$year <= 1980),]
x <- data$year
y <- data$population
A_q <- matrix(c(rep(1, length(x)), x, x^2), nr = length(x), nc = 3)
```

Normal equations failure

Recall that using the normal equations on this matrix failed because $A^T A$ is computationally singular. We can see this in R:

```
ATA <- t(A_q) %*% A_q
try(solve(ATA) %*% t(A_q) %*% y)

## Error in solve.default(ATA) :
##   system is computationally singular: reciprocal condition number = 9.166
```

However, the QR decomposition avoids forming $A^T A$, preserving numerical stability.

Solving with QR decomposition

We can compute the QR decomposition using R's built-in `qr` function, extract the Q and R components, and then use `backsolve` to solve the upper triangular system $R\mathbf{x} = Q^T \mathbf{b}$.

```
# Compute the QR decomposition of A_q
QR <- qr(A_q)
Q <- qr.Q(QR)
R <- qr.R(QR)

# Solve Rx = Q^T y
sol_qr <- backsolve(R, t(Q) %*% y)
sol_qr

##           [,1]
## [1,]  5.133645e+09
## [2,] -5.513814e+06
## [3,]  1.481439e+03
```

Direct solution with `qr.solve`

The QR decomposition successfully finds the coefficients for the unshifted data!

Alternatively, R provides the function `qr.solve` which directly solves the least squares problem using the QR decomposition under the hood:

```
sol_qr_direct <- qr.solve(A_q, y)
sol_qr_direct

## [1] 5.133645e+09 -5.513814e+06 1.481439e+03
```

This demonstrates the practical advantage of using orthogonal matrix factorisations in data science applications over the normal equations.

The QR decomposition

The SVD (singular values decomposition)

The SVD (singular values decomposition)

Singular values

The SVD

An application of the SVD – Image compression

Computing the SVD

Applications of the SVD – Least squares

Summary – Least squares methods

Matrix factorisations (continued)

The singular value decomposition (known mostly by its acronym, SVD) is yet another type of factorisation/decomposition..

Singular values

Definition 73 (Singular value)

Let $A \in \mathcal{M}_{np}(\mathbb{R})$. The **singular values** of A are the real numbers

$$\sigma_1 \geq \sigma_2 \geq \cdots \sigma_p \geq 0$$

that are the square roots of the eigenvalues of $A^T A$

Singular values are real and nonnegative?

Recall that $\forall A \in \mathcal{M}_{np}$, $A^T A$ is symmetric

Theorem 74

Let $A \in \mathcal{M}_n(\mathbb{R})$ be symmetric. Then all eigenvalues of A are real

Proof of Theorem 74

$A \in \mathcal{M}_n(\mathbb{R})$ symmetric and $(\lambda, \mathbf{v})$ eigenpair of A , i.e, $A\mathbf{v} = \lambda\mathbf{v}$. Taking the complex conjugate, $\overline{A\mathbf{v}} = \overline{\lambda\mathbf{v}}$

Since $A \in \mathcal{M}_n(\mathbb{R})$, $\overline{A} = A$ $(z = \bar{z} \iff z \in \mathbb{R})$

So

$$A\bar{\mathbf{v}} = \overline{A\mathbf{v}} = \overline{\lambda\mathbf{v}} = \bar{\lambda}\bar{\mathbf{v}}$$

i.e., if $(\lambda, \mathbf{v})$ eigenpair, $(\bar{\lambda}, \bar{\mathbf{v}})$ also eigenpair

Proof of Theorem 74 (cont.)

Still assuming $A \in \mathcal{M}_n(\mathbb{R})$ symmetric and $(\lambda, \mathbf{v})$ eigenpair of A and using what we just proved (that $(\bar{\lambda}, \bar{\mathbf{v}})$ also eigenpair), take transposes

$$\begin{aligned} A\bar{\mathbf{v}} = \bar{\lambda}\bar{\mathbf{v}} &\iff (A\bar{\mathbf{v}})^T = (\bar{\lambda}\bar{\mathbf{v}})^T \\ &\iff \bar{\mathbf{v}}^T A^T = \bar{\lambda}\bar{\mathbf{v}}^T \\ &\iff \bar{\mathbf{v}}^T A = \bar{\lambda}\bar{\mathbf{v}}^T \quad [A \text{ symmetric}] \end{aligned}$$

Let us now compute $\lambda(\bar{\mathbf{v}} \bullet \mathbf{v})$. We have

$$\begin{aligned} \lambda(\bar{\mathbf{v}} \bullet \mathbf{v}) &= \lambda \bar{\mathbf{v}}^T \mathbf{v} = \bar{\mathbf{v}}^T (\lambda \mathbf{v}) \\ &= \bar{\mathbf{v}}^T (A\mathbf{v}) = (\bar{\mathbf{v}}^T A) \mathbf{v} \\ &= (\bar{\lambda} \bar{\mathbf{v}}^T) \mathbf{v} = \bar{\lambda}(\bar{\mathbf{v}} \bullet \mathbf{v}) \\ &\iff (\lambda - \bar{\lambda})(\bar{\mathbf{v}} \bullet \mathbf{v}) = 0 \end{aligned}$$

Proof of Theorem 74 (end)

We have shown

$$(\lambda - \bar{\lambda})(\bar{\mathbf{v}} \bullet \mathbf{v}) = 0$$

Let

$$\mathbf{v} = \begin{pmatrix} a_1 + ib_1 \\ \vdots \\ a_n + ib_n \end{pmatrix}$$

Then

$$\bar{\mathbf{v}} = \begin{pmatrix} a_1 - ib_1 \\ \vdots \\ a_n - ib_n \end{pmatrix}$$

So

$$\bar{\mathbf{v}} \bullet \mathbf{v} = (a_1^2 + b_1^2) + \cdots + (a_n^2 + b_n^2)$$

But $\mathbf{v}$ eigenvector is $\neq \mathbf{0}$, so $\bar{\mathbf{v}} \bullet \mathbf{v} \neq 0$, so

$$(\lambda - \bar{\lambda})(\bar{\mathbf{v}} \bullet \mathbf{v}) = 0 \iff \lambda - \bar{\lambda} = 0 \iff \lambda = \bar{\lambda} \iff \lambda \in \mathbb{R} \quad \square$$

Eigenvalues of $A^T A$ are real and nonnegative

Theorem 75

For $A \in \mathcal{M}_{np}(\mathbb{R})$, the eigenvalues of $A^T A$ (and AA^T) are real and nonnegative

Proof of Theorem 75

We know that for $A \in \mathcal{M}_{np}$, $A^T A$ symmetric and from Theorem 74, if $A \in \mathcal{M}_{np}(\mathbb{R})$, then $A^T A$ is symmetric, real and with real eigenvalues

Let $(\lambda, \mathbf{v})$ be an eigenpair of $A^T A$, with $\mathbf{v}$ chosen so that $\|\mathbf{v}\| = 1$

Norms are functions $V \rightarrow \mathbb{R}_+$, so $\|A\mathbf{v}\|$ and $\|A\mathbf{v}\|^2$ are ≥ 0 and thus

$$\begin{aligned} 0 \leq \|A\mathbf{v}\|^2 &= (A\mathbf{v}) \bullet (A\mathbf{v}) = (A\mathbf{v})^T (A\mathbf{v}) \\ &= \mathbf{v}^T A^T A \mathbf{v} = \mathbf{v}^T (A^T A \mathbf{v}) = \mathbf{v}^T (\lambda \mathbf{v}) \\ &= \lambda (\mathbf{v}^T \mathbf{v}) = \lambda (\mathbf{v} \bullet \mathbf{v}) = \lambda \|\mathbf{v}\|^2 \\ &= \lambda \end{aligned}$$

Nonzero eigenvalues of $A^T A$ and AA^T

Theorem 76

For $A \in \mathcal{M}_{np}(\mathbb{R})$, the nonzero eigenvalues of $A^T A$ and AA^T are the same

Proof.

Let $(\lambda, \mathbf{v})$ be an eigenpair of $A^T A$ with $\lambda \neq 0$. Then $\mathbf{v} \neq \mathbf{0}$ and

$$A^T A \mathbf{v} = \lambda \mathbf{v} \neq \mathbf{0}$$

Left multiply by A

$$AA^T A \mathbf{v} = \lambda A \mathbf{v}$$

Let $\mathbf{w} = A \mathbf{v}$, we thus have $AA^T \mathbf{w} = \lambda \mathbf{w}$; in other words, $A \mathbf{v}$ is an eigenvector of AA^T corresponding to the (nonzero) eigenvalue λ

The reverse works the same way..



The SVD (singular values decomposition)

Singular values

The SVD

An application of the SVD – Image compression

Computing the SVD

Applications of the SVD – Least squares

Summary – Least squares methods

The singular value decomposition (SVD)

Theorem 77 (SVD)

$A \in \mathcal{M}_{np}$ with singular values $\sigma_1 \geq \dots \geq \sigma_r > 0$ and $\sigma_{r+1} = \dots = \sigma_p = 0$

Then there exists $U \in \mathcal{M}_n$ orthogonal, $V \in \mathcal{M}_p$ orthogonal and a block matrix $\Sigma \in \mathcal{M}_{np}$ taking the form

$$\Sigma = \begin{pmatrix} D & \mathbf{0}_{r,p-r} \\ \mathbf{0}_{n-r,r} & \mathbf{0}_{n-r,p-r} \end{pmatrix}$$

where

$$D = \text{diag}(\sigma_1, \dots, \sigma_r) \in \mathcal{M}_r$$

such that

$$A = U\Sigma V^T$$

Definition 78

We call a factorisation as in Theorem 77 the **singular value decomposition** of A . The columns of U and V are, respectively, the **left** and **right singular vectors** of A

U and V^T are *rotation* or *reflection* matrices, Σ is a *scaling* matrix

$U \in \mathcal{M}_n$ orthogonal matrix with columns the eigenvectors of AA^T

$V \in \mathcal{M}_p$ orthogonal matrix with columns the eigenvectors of $A^T A$

Outer product form of the SVD

Theorem 79 (Outer product form of the SVD)

$A \in \mathcal{M}_{np}$ with singular values $\sigma_1 \geq \dots \geq \sigma_r > 0$ and $\sigma_{r+1} = \dots = \sigma_p = 0$, $\mathbf{u}_1, \dots, \mathbf{u}_r$ and $\mathbf{v}_1, \dots, \mathbf{v}_r$, respectively, left and right singular vectors of A corresponding to these singular values

Then

$$A = \sigma_1 \mathbf{u}_1 \mathbf{v}_1^T + \dots + \sigma_r \mathbf{u}_r \mathbf{v}_r^T \quad (2)$$

The SVD (singular values decomposition)

Singular values

The SVD

An application of the SVD – Image compression

Computing the SVD

Applications of the SVD – Least squares

Summary – Least squares methods

Applications of the SVD

Many applications of the SVD, both theoretical and practical..

1. Obtaining a unique solutions to least squares when $A^T A$ singular
2. Image compression

Compressing images

Consider an image (for simplicity, assume in shades of grey). This can be stored in a matrix $A \in \mathcal{M}_{np}$

Take the SVD of A . Then the small singular values carry information about the regions with little variation and can perhaps be omitted, whereas the large singular values carry information about more “dynamic” regions of the image

Suppose A has r nonzero singular values. For $k \leq r$, let

$$A_k = \sigma_1 \mathbf{u}_1 \mathbf{v}_1^T + \cdots + \sigma_k \mathbf{u}_k \mathbf{v}_k^T$$

For $k = r$ we get the usual outer product form (2) in Theorem 79

Load the image using `bmp::read.bmp`

```
my_image = bmp::read.bmp("../CODE/Julien_and_friend_1000x800.bmp")
my_image_g = pixmap::pixmapGrey(my_image)
my_image_g

## Pixmap image
##   Type           : pixmapGrey
##   Size            : 800x1000
##   Resolution      : 1x1
##   Bounding box    : 0 0 1000 800
```



Using built-in functions

We can also use the built-in function `svd` to compute the SVD of M

```
M.svd = svd(M)
```

The results are stored in a list with components u , d and v

Make function to recreate an image from the SVD

Given the SVD `im` of an image and a number of singular values to keep `n`, we can recreate the image using the function `compress_image`

We output the new image, but also, the amount of information required to encode this new image, as a percentage of the original image size

```

compress_image = function(im, n) {
  if (n > length(im$d)) {
    # Check that we gave a value of n within range, otherwise
    # just set to the max
    n = length(im$d)
  }
  d_tmp = im$d[1:n]
  u_tmp = im$u[,1:n]
  v_tmp = im$v[,1:n]
  # Store the results in a list (so we can return other information)
  out = list()
  # First, compute the resulting image
  out$img = mat.or.vec(nr = dim(im$u)[1], nc = dim(im$v)[1])
  for (i in 1:n) {
    out$img = out$img + d_tmp[i] * u_tmp[,i] %*% t(v_tmp[,i])
  }
}

```

```

# Values of "colours" must be in [0,1], so shift and rescale
if (min(min(out$img)) < 0 ) {
    out$img = out$img - min(min(out$img))
}
out$img = out$img / max(max(out$img))
# Store some information: # points needed and % of the original
out$nb_pixels_original = dim(im$u)[1] * dim(im$v)[2]
out$nb_pixels_compressed = length(d_tmp) +
    dim(u_tmp)[1]*dim(u_tmp)[2] +
    dim(v_tmp)[1]*dim(v_tmp)[2]
out$pct_of_original = out$nb_pixels_compressed /
    out$nb_pixels_original * 100
# Return the result
return(out)
}

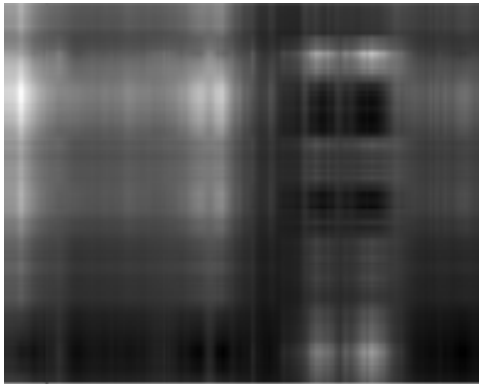
```

Recreating the image

We can now recreate the image using the function `compress_image`

```
new_image = my_image_g
M.svd = svd(M)
M_tmp = compress_image(M.svd, 2)
new_image@grey = M_tmp$img
plot(new_image)
```

Using $n = 2$ singular values



Uses 0.56% of the original information

Using $n = 5$ singular values



Uses 1.41% of the original information

Using $n = 10$ singular values



Uses 2.81% of the original information

Using $n = 20$ singular values



Uses 5.63% of the original information

Using $n = 50$ singular values



Uses 14.07% of the original information

The SVD (singular values decomposition)

Singular values

The SVD

An application of the SVD – Image compression

Computing the SVD

Applications of the SVD – Least squares

Summary – Least squares methods

Computing the SVD (case of $\neq$ eigenvalues)

To compute the SVD, we use the following result

Theorem 80

Let $A \in \mathcal{M}_n$ symmetric, $(\lambda_1, \mathbf{u}_1)$ and $(\lambda_2, \mathbf{u}_2)$ be eigenpairs, $\lambda_1 \neq \lambda_2$. Then $\mathbf{u}_1 \bullet \mathbf{u}_2 = 0$

Proof of Theorem 80

$A \in \mathcal{M}_n$ symmetric, $(\lambda_1, \mathbf{u}_1)$ and $(\lambda_2, \mathbf{u}_2)$ eigenpairs with $\lambda_1 \neq \lambda_2$

$$\begin{aligned}\lambda_1(\mathbf{v}_1 \bullet \mathbf{v}_2) &= (\lambda_1 \mathbf{v}_1) \bullet \mathbf{v}_2 \\ &= A\mathbf{v}_1 \bullet \mathbf{v}_2 \\ &= (A\mathbf{v}_1)^T \mathbf{v}_2 \\ &= \mathbf{v}_1^T A^T \mathbf{v}_2 \\ &= \mathbf{v}_1^T (A\mathbf{v}_2) \quad [A \text{ symmetric so } A^T = A] \\ &= \mathbf{v}_1^T (\lambda_2 \mathbf{v}_2) \\ &= \lambda_2 (\mathbf{v}_1^T \mathbf{v}_2) \\ &= \lambda_2 (\mathbf{v}_1 \bullet \mathbf{v}_2)\end{aligned}$$

So $(\lambda_1 - \lambda_2)(\mathbf{v}_1 \bullet \mathbf{v}_2) = 0$. But $\lambda_1 \neq \lambda_2$, so $\mathbf{v}_1 \bullet \mathbf{v}_2 = 0$



Computing the SVD (case of $\neq$ eigenvalues)

If all eigenvalues of $A^T A$ (or AA^T) are distinct, we can use Theorem 80

1. Compute $A^T A \in \mathcal{M}_p$
2. Compute eigenvalues $\lambda_1, \dots, \lambda_p$ of $A^T A$; order them as $\lambda_1 > \dots > \lambda_p \geq 0$ ($>$ not $\geq$ since $\neq$)
3. Compute singular values $\sigma_1 = \sqrt{\lambda_1}, \dots, \sigma_p = \sqrt{\lambda_p}$
4. Diagonal matrix D in Σ is either in $\mathcal{M}_p$ (if $\sigma_p > 0$) or in $\mathcal{M}_{p-1}$ (if $\sigma_p = 0$)

5. Since eigenvalues are distinct, Theorem 80 $\implies$ eigenvectors are orthogonal set. Compute these eigenvectors in the same order as the eigenvalues
6. Normalise them and use them to make the matrix V , i.e., $V = [\mathbf{v}_1 \cdots \mathbf{v}_p]$
7. To find the $\mathbf{u}_i$, compute, for $i = 1, \dots, r$,

$$\mathbf{u}_i = \frac{1}{\sigma_i} A \mathbf{v}_i$$

and ensure that $\|\mathbf{u}_i\| = 1$

Why do we compute $\mathbf{u}_i$ this way?

Recall the definition of the SVD:

$$A = U\Sigma V^T$$

Right-multiply by V ; since V is an orthogonal matrix ($V^T V = VV^T = \mathbb{I}$)

$$AV = U\Sigma V^T V = U\Sigma$$

Looking at the i -th column of both sides of this equation:

- ▶ The i -th column of AV is $A\mathbf{v}_i$
- ▶ The i -th column of $U\Sigma$ is $\sigma_i \mathbf{u}_i$

Therefore, we have $A\mathbf{v}_i = \sigma_i \mathbf{u}_i$, which yields for $\sigma_i > 0$:

$$\mathbf{u}_i = \frac{1}{\sigma_i} A\mathbf{v}_i$$

Example 1: Square matrix

Let $A = \begin{pmatrix} 2 & 2 \\ -1 & 1 \end{pmatrix}$.

1. Compute $A^T A = \begin{pmatrix} 2 & -1 \\ 2 & 1 \end{pmatrix} \begin{pmatrix} 2 & 2 \\ -1 & 1 \end{pmatrix} = \begin{pmatrix} 5 & 3 \\ 3 & 5 \end{pmatrix}$
2. Eigenvalues of $A^T A$: characteristic equation is $(5 - \lambda)^2 - 9 = 0 \implies \lambda = 8, 2$.
So $\lambda_1 = 8, \lambda_2 = 2$.
3. Singular values: $\sigma_1 = \sqrt{8} = 2\sqrt{2}, \sigma_2 = \sqrt{2}$. Matrix $\Sigma = \begin{pmatrix} 2\sqrt{2} & 0 \\ 0 & \sqrt{2} \end{pmatrix}$.

Example 1: Square matrix (continued)

4. Eigenvectors of $A^T A$:

- ▶ For $\lambda_1 = 8$: null space of $\begin{pmatrix} -3 & 3 \\ 3 & -3 \end{pmatrix}$, giving $\mathbf{v}_1 = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix}$
- ▶ For $\lambda_2 = 2$: null space of $\begin{pmatrix} 3 & 3 \\ 3 & 3 \end{pmatrix}$, giving $\mathbf{v}_2 = \frac{1}{\sqrt{2}} \begin{pmatrix} -1 \\ 1 \end{pmatrix}$

$$\text{So } V = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix}.$$

5. Compute the $\mathbf{u}_i$:

- ▶ $\mathbf{u}_1 = \frac{1}{\sigma_1} A \mathbf{v}_1 = \frac{1}{2\sqrt{2}} \begin{pmatrix} 2 & 2 \\ -1 & 1 \end{pmatrix} \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \frac{1}{4} \begin{pmatrix} 4 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$
- ▶ $\mathbf{u}_2 = \frac{1}{\sigma_2} A \mathbf{v}_2 = \frac{1}{\sqrt{2}} \begin{pmatrix} 2 & 2 \\ -1 & 1 \end{pmatrix} \frac{1}{\sqrt{2}} \begin{pmatrix} -1 \\ 1 \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 0 \\ 2 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$

$$\text{So } U = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

Example 2: Nonsquare matrix

Let $A = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \end{pmatrix}$.

1. Compute $A^T A = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix}$
2. Eigenvalues of $A^T A$ are clearly $\lambda_1 = 2, \lambda_2 = 1$.
3. Singular values: $\sigma_1 = \sqrt{2}, \sigma_2 = 1$. The matrix Σ is 3×2 , so $\Sigma = \begin{pmatrix} \sqrt{2} & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix}$.
4. Eigenvectors of $A^T A$ are the standard basis vectors: $\mathbf{v}_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \mathbf{v}_2 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$.

So $V = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$.

Example 2: Nonsquare matrix (continued)

5. Compute the $\mathbf{u}_i$ for $i = 1, 2$:

$$\blacktriangleright \mathbf{u}_1 = \frac{1}{\sigma_1} A \mathbf{v}_1 = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}$$

$$\blacktriangleright \mathbf{u}_2 = \frac{1}{\sigma_2} A \mathbf{v}_2 = \frac{1}{1} \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$$

6. Find $\mathbf{u}_3$ to complete the orthonormal basis U of $\mathcal{R}^3$. We need $\mathbf{u}_3$ orthogonal to $\mathbf{u}_1$ and $\mathbf{u}_2$, with $\|\mathbf{u}_3\| = 1$. By inspection, $\mathbf{u}_3 = \frac{1}{\sqrt{2}} \begin{pmatrix} -1 \\ 0 \\ 1 \end{pmatrix}$ works.

7. Finally, $U = \begin{pmatrix} \frac{1}{\sqrt{2}} & 0 & -\frac{1}{\sqrt{2}} \\ 0 & 1 & 0 \\ \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} \end{pmatrix}$.

Return to image compression – Computations “by hand”

```
M = my_image_g@grey
MTM = t(M) %*% M
# Ensure matrix is symmetric
MTM = (MTM+t(MTM))/2
ev = eigen(MTM)
```

Given the size and nature of the entries, the matrix $M^T M$ is symmetric only to $1e-5$ precision, so we use a little trick to make it symmetric no matter what: take the average of $M^T M$ and its transpose MM^T

Which version of the algorithm to use?

Make zero the eigenvalues that are close to zero (200 out of 1000)

```
ev$values = ev$values*(ev$values>1e-10)
```

Can we use the algorithm for all eigenvalues being distinct or do we have repeated ones?

```
any(duplicated(ev$values[ev$values>1e-10]))
```

```
## [1] FALSE
```

So we can use the standard algorithm

Computing the SVD

```
idx_positive_ev = which(ev$values>1e-10)
sv = sqrt(ev$values[idx_positive_ev])
```

Computing the SVD

Then $D = \text{diag}(\sigma_1, \dots, \sigma_r)$, V is the matrix of normalised eigenvectors in the same order as the σ_i and for $i = 1, \dots, r$

$$\mathbf{u}_i = \frac{1}{\sigma_i} A \mathbf{v}_i$$

ensuring that $\|\mathbf{u}_i\| = 1$

```
D = diag(sv)
V = ev$eigenvectors[, idx_positive_ev]
# Not needed with R, other languages might not normalise
c1 = sqrt(colSums(V^2))
for (i in 1:dim(V)[2]) {
  V[,i] = V[,i]/c1[i]
}
```

Computing the SVD

Finally, we compute the $\mathbf{u}_i$'s

```
U = M %*% V %*% diag(1/sv)
r = length(sv)
im = list(u=U, d=sv, v=V)
```

Computing the SVD (case where some eigenvalues are $=$)

1. Compute $A^T A \in \mathcal{M}_p$
2. Compute eigenvalues $\lambda_1, \dots, \lambda_p$ of $A^T A$; order them as $\lambda_1 \geq \dots \geq \lambda_p \geq 0$
3. Compute singular values $\sigma_1 = \sqrt{\lambda_1}, \dots, \sigma_p = \sqrt{\lambda_p}$, with $r \leq p$ the index of the last positive singular value
4. For eigenvalues that are distinct, proceed as before
5. For eigenvalues with multiplicity > 1 , we need to ensure that the resulting eigenvectors are LI *and* orthogonal

Dealing with eigenvalues with multiplicity > 1

When an eigenvalue has (algebraic) multiplicity > 1 , e.g., characteristic polynomial contains a factor like $(\lambda - 2)^2$, things can become a little bit more complicated

The proper way to deal with this involves the so-called Jordan Normal Form (JNF, another matrix decomposition – for square matrices)

In short: not all square matrices are diagonalisable, but all square matrices admit a JNF

Sometimes, we can find several LI eigenvectors associated to the same eigenvalue. Check this. If not, need to use the following

Definition 81 (Generalised eigenvectors)

The vector $\mathbf{x} \neq \mathbf{0}$ is a **generalized eigenvector** of rank m of $A \in \mathcal{M}_n$ corresponding to eigenvalue λ if

$$(A - \lambda \mathbb{I})^m \mathbf{x} = \mathbf{0}$$

but

$$(A - \lambda \mathbb{I})^{m-1} \mathbf{x} \neq \mathbf{0}$$

Procedure for generalised eigenvectors

$A \in \mathcal{M}_n$ and assume λ eigenvalue with algebraic multiplicity k

Find $\mathbf{v}_1$, "classic" eigenvector, i.e., $\mathbf{v}_1 \neq \mathbf{0}$ s.t. $(A - \lambda\mathbb{I})\mathbf{v}_1 = \mathbf{0}$

Find generalised eigenvector $\mathbf{v}_2$ of rank 2 by solving for $\mathbf{v}_2 \neq \mathbf{0}$,

$$(A - \lambda\mathbb{I})\mathbf{v}_2 = \mathbf{v}_1$$

...

Find generalised eigenvector $\mathbf{v}_k$ of rank k by solving for $\mathbf{v}_k \neq \mathbf{0}$,

$$(A - \lambda\mathbb{I})\mathbf{v}_k = \mathbf{v}_{k-1}$$

Then $\{\mathbf{v}_1, \dots, \mathbf{v}_k\}$ LI

Back to the normal procedure

With the LI eigenvectors $\{\mathbf{v}_1, \dots, \mathbf{v}_k\}$ corresponding to λ

Apply Gram-Schmidt to get orthogonal set

For all eigenvalues with multiplicity > 1 , check that you either have LI eigenvectors or do what we just did

When you are done, be back on your merry way to step 6 in the case where eigenvalues are all $\neq$

I am caricaturing a little here: there can be cases that do not work exactly like this, but this is general enough..

Example 3: Repeated eigenvalues

$$\text{Let } A = \begin{pmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \\ -1 & 0 & 1 \end{pmatrix}.$$

1. Compute $A^T A = \begin{pmatrix} 1 & 0 & -1 \\ -1 & 1 & 0 \\ 0 & -1 & 1 \end{pmatrix} \begin{pmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \\ -1 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{pmatrix}$
2. Eigenvalues of $A^T A$: the characteristic polynomial is $-\lambda^3 + 6\lambda^2 - 9\lambda = 0$, which gives $\lambda_1 = 3, \lambda_2 = 3, \lambda_3 = 0$.
3. Singular values: $\sigma_1 = \sqrt{3}, \sigma_2 = \sqrt{3}, \sigma_3 = 0$.

Example 3: Repeated eigenvalues (continued)

4. Eigenvectors of $A^T A$:

► For $\lambda_3 = 0$: null space of $A^T A$, giving $\mathbf{v}_3 = \frac{1}{\sqrt{3}} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$

► For $\lambda_1 = \lambda_2 = 3$: we need to find the null space of

$$A^T A - 3\mathbb{I} = \begin{pmatrix} -1 & -1 & -1 \\ -1 & -1 & -1 \\ -1 & -1 & -1 \end{pmatrix}. \text{ The eigenspace is the plane } x + y + z = 0.$$

We must choose two **orthogonal** eigenvectors in this plane.

Let $\mathbf{x}_1 = \begin{pmatrix} 1 \\ -1 \\ 0 \end{pmatrix}$. To find a second vector $\mathbf{x}_2$ orthogonal to $\mathbf{x}_1$ using

Gram-Schmidt, we start with another vector in the plane, e.g., $\mathbf{y}_2 = \begin{pmatrix} 1 \\ 0 \\ -1 \end{pmatrix}$.

We compute $\mathbf{x}_2 = \mathbf{y}_2 - \text{proj}_{\mathbf{x}_1}(\mathbf{y}_2) = \begin{pmatrix} 1/2 \\ 1/2 \\ -1 \end{pmatrix}$, which we can scale to $\begin{pmatrix} 1 \\ 1 \\ -2 \end{pmatrix}$.

Example 3: Repeated eigenvalues (continued)

5. Compute the $\mathbf{u}_i$ for $i = 1, 2$:

$$\blacktriangleright \mathbf{u}_1 = \frac{1}{\sigma_1} \mathbf{A} \mathbf{v}_1 = \frac{1}{\sqrt{3}} \begin{pmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \\ -1 & 0 & 1 \end{pmatrix} \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -1 \\ 0 \end{pmatrix} = \frac{1}{\sqrt{6}} \begin{pmatrix} 2 \\ -1 \\ -1 \end{pmatrix}$$

$$\blacktriangleright \mathbf{u}_2 = \frac{1}{\sigma_2} \mathbf{A} \mathbf{v}_2 = \frac{1}{\sqrt{3}} \begin{pmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \\ -1 & 0 & 1 \end{pmatrix} \frac{1}{\sqrt{6}} \begin{pmatrix} 1 \\ 1 \\ -2 \end{pmatrix} = \frac{1}{\sqrt{18}} \begin{pmatrix} 0 \\ 3 \\ -3 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 0 \\ 1 \\ -1 \end{pmatrix}$$

6. To complete the orthonormal basis U of $\mathcal{R}^3$, we find $\mathbf{u}_3$ orthogonal to $\mathbf{u}_1$ and $\mathbf{u}_2$. By inspection (or cross product), $\mathbf{u}_3 = \frac{1}{\sqrt{3}} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$.

7. Finally, $U = \begin{pmatrix} \frac{2}{\sqrt{6}} & 0 & \frac{1}{\sqrt{3}} \\ -\frac{1}{\sqrt{6}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{3}} \\ -\frac{1}{\sqrt{6}} & -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{3}} \end{pmatrix}$.

The SVD (singular values decomposition)

Singular values

The SVD

An application of the SVD – Image compression

Computing the SVD

Applications of the SVD – Least squares

Summary – Least squares methods

Pseudoinverse of a matrix

Definition 82 (Pseudoinverse)

$A = U\Sigma V^T$ an SVD for $A \in \mathcal{M}_{np}$, where

$$\Sigma = \begin{pmatrix} D & 0 \\ 0 & 0 \end{pmatrix}, \text{ with } D = \text{diag}(\sigma_1, \dots, \sigma_r)$$

(D contains the nonzero singular values of A ordered as usual)

The **pseudoinverse** (or **Moore-Penrose inverse**) of A is $A^+ \in \mathcal{M}_{pn}$ given by

$$A^+ = V\Sigma^+ U^T$$

with

$$\Sigma^+ = \begin{pmatrix} D^{-1} & 0 \\ 0 & 0 \end{pmatrix} \in \mathcal{M}_{pn}$$

Properties of the pseudoinverse

The pseudoinverse A^+ generalises the concept of matrix inverse and satisfies the following properties (known as the Moore-Penrose conditions):

1. $AA^+A = A$
2. $A^+AA^+ = A^+$
3. $(AA^+)^T = AA^+$
4. $(A^+A)^T = A^+A$

Other useful properties:

- ▶ If A is square and invertible, then $A^+ = A^{-1}$
- ▶ $(A^+)^+ = A$
- ▶ $(A^T)^+ = (A^+)^T$
- ▶ If A has full column rank, then $A^+ = (A^T A)^{-1} A^T$
- ▶ If A has full row rank, then $A^+ = A^T (A A^T)^{-1}$

Least squares revisited

Theorem 83

Let $A \in \mathcal{M}_{np}$, $\mathbf{x} \in \mathbb{R}^p$ and $\mathbf{b} \in \mathbb{R}^n$. The least squares problem $A\mathbf{x} = \mathbf{b}$ has a unique least squares solution $\tilde{\mathbf{x}}$ of minimal length (closest to the origin) given by

$$\tilde{\mathbf{x}} = A^+ \mathbf{b}$$

where A^+ is the pseudoinverse of A

What does “minimal length” mean?

When A has *full column rank* (i.e., its columns are linearly independent), the least squares solution is unique:

$$\tilde{\mathbf{x}} = (A^T A)^{-1} A^T \mathbf{b}$$

However, if the columns of A are *linearly dependent*, the matrix $A^T A$ is not invertible. In this case, there are **infinitely many** least squares solutions that minimise $\|A\mathbf{x} - \mathbf{b}\|$

Among all these possible solutions, the pseudoinverse A^+ selects the unique solution $\tilde{\mathbf{x}} = A^+ \mathbf{b}$ that has the **smallest Euclidean norm** $\|\tilde{\mathbf{x}}\|$

Geometrically, it is the solution closest to the origin in $\mathbb{R}^p$

Example: Computing the pseudoinverse

Let $A = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$. The SVD of A is $U\Sigma V^T$ with:

$$U = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \quad \Sigma = \begin{pmatrix} 2 & 0 \\ 0 & 0 \end{pmatrix}, \quad V = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

To find A^+ , we form Σ^+ by taking the reciprocal of the non-zero singular values (leaving zeros as zero):

$$\Sigma^+ = \begin{pmatrix} 1/2 & 0 \\ 0 & 0 \end{pmatrix}$$

Then the pseudoinverse is $A^+ = V\Sigma^+U^T$:

$$A^+ = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} 1/2 & 0 \\ 0 & 0 \end{pmatrix} \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}^T = \frac{1}{4} \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$$

Example: Least squares using the pseudoinverse

Consider $A\mathbf{x} = \mathbf{b}$ where $A = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$ and $\mathbf{b} = \begin{pmatrix} 3 \\ 1 \end{pmatrix}$

The system $\begin{pmatrix} x_1 + x_2 \\ x_1 + x_2 \end{pmatrix} = \begin{pmatrix} 3 \\ 1 \end{pmatrix}$ has no exact solution

The least squares approach minimises the error

$\|A\mathbf{x} - \mathbf{b}\|^2 = (x_1 + x_2 - 3)^2 + (x_1 + x_2 - 1)^2$. The minimum error is achieved when $x_1 + x_2 = 2$

Any vector $\mathbf{x} = \begin{pmatrix} x_1 \\ 2 - x_1 \end{pmatrix}$ is a least squares solution (e.g., $\begin{pmatrix} 2 \\ 0 \end{pmatrix}$, $\begin{pmatrix} 0 \\ 2 \end{pmatrix}$)

The **minimal length** solution is given by the pseudoinverse:

$$\tilde{\mathbf{x}} = A^+ \mathbf{b} = \frac{1}{4} \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} 3 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

which is indeed the point on the line $x_1 + x_2 = 2$ closest to the origin

The SVD (singular values decomposition)

Singular values

The SVD

An application of the SVD – Image compression

Computing the SVD

Applications of the SVD – Least squares

Summary – Least squares methods

The least squares problem

Problem Statement:

Given a system $A\mathbf{x} = \mathbf{b}$ where $A \in \mathcal{M}_{np}$, $\mathbf{x} \in \mathbb{R}^p$, $\mathbf{b} \in \mathbb{R}^n$ (typically $n > p$), find $\tilde{\mathbf{x}}$ that minimizes

$$\|\mathbf{b} - A\mathbf{x}\|^2 = \sum_{i=1}^n (b_i - \sum_{j=1}^p A_{ij}x_j)^2$$

Geometric interpretation: Find the vector $A\tilde{\mathbf{x}}$ in the column space of A that is closest to $\mathbf{b}$

Solution: $A\tilde{\mathbf{x}} = \text{proj}_{\text{col}(A)}(\mathbf{b})$

Method 1: Normal equations

The normal equations:

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

When this works:

- ▶ Always has at least one solution
- ▶ Any solution $\tilde{\mathbf{x}}$ to the normal equations is a least squares solution

Computational issues:

- ▶ Forming $A^T A$ can be numerically unstable
- ▶ Condition number of $A^T A$ is the square of the condition number of A
- ▶ Still useful for theoretical analysis

Method 2: when A Has linearly independent columns

Condition: $A \in \mathcal{M}_{np}$ has linearly independent columns

Then: $A^T A$ is invertible and the least squares solution is **unique**

$$\tilde{\mathbf{x}} = (A^T A)^{-1} A^T \mathbf{b}$$

Properties:

- ▶ $A^T A \in \mathcal{M}_p$ is square, symmetric, and positive definite
- ▶ $(A^T A)^{-1} A^T$ is called the *left pseudoinverse* of A
- ▶ This gives the unique least squares solution

Drawback: Computing $(A^T A)^{-1}$ directly can be numerically unstable

Method 3: QR factorization

QR Factorization: If $A \in \mathcal{M}_{np}$ has linearly independent columns, then

$$A = QR$$

where $Q \in \mathcal{M}_{np}$ has orthonormal columns and $R \in \mathcal{M}_p$ is upper triangular and nonsingular

Least squares solution:

$$\tilde{\mathbf{x}} = R^{-1} Q^T \mathbf{b}$$

Advantages:

- ▶ More numerically stable than forming $A^T A$
- ▶ R is upper triangular $\Rightarrow$ solving $R\tilde{\mathbf{x}} = Q^T \mathbf{b}$ by back substitution
- ▶ Condition number of R equals condition number of A
- ▶ Gram-Schmidt or Householder reflections can compute QR factorization

Method 4: Singular Value Decomposition (SVD)

SVD: For any $A \in \mathcal{M}_{np}$,

$$A = U\Sigma V^T$$

where $U \in \mathcal{M}_n$ orthogonal, $V \in \mathcal{M}_p$ orthogonal, $\Sigma \in \mathcal{M}_{np}$ with $\Sigma_{ii} = \sigma_i \geq 0$ (singular values)

Pseudoinverse: $A^+ = V\Sigma^+U^T$ where Σ^+ has $(\Sigma^+)_{ii} = 1/\sigma_i$ if $\sigma_i > 0$, else 0

Least squares solution:

$$\tilde{\mathbf{x}} = A^+ \mathbf{b}$$

Key advantages:

- ▶ Works for *any* matrix A (even when columns are linearly dependent)
- ▶ Gives the solution of *minimal length* when multiple solutions exist
- ▶ Most numerically stable method
- ▶ Reveals the rank of A through the number of non-zero singular values

When to use which method

Method	When to use	Advantages/Drawbacks
Normal equations	Theory, small problems	Simple, but unstable
$(A^T A)^{-1} A^T$	A has LI columns	Explicit formula, unstable
QR Factorization	A has LI columns	Stable, efficient
SVD	Any A , rank-deficient	Most stable, handles all cases

Use QR for well-conditioned problems with LI columns, SVD for rank-deficient or ill-conditioned problems