Numerical Matrix Analysis

Notes #6 — The QR-Factorization and Least Squares Problems: Orthogonality and Projections

Peter Blomgren
⟨blomgren@sdsu.edu⟩

Department of Mathematics and Statistics
Dynamical Systems Group
Computational Sciences Research Center
San Diego State University
San Diego, CA 92182-7720

http://terminus.sdsu.edu/

Spring 2024
(Revised: February 20, 2024)

Outline

Student Learning Targets, and Objectives

Target The QR-Factorization
- Objective How to compute using the Gram-Schmidt Orthogonalization Method

Target Building Blocks
- Objective Projectors, Idempotent Matrices, Complementary Projectors
- Objective Characterization of the SVD using Orthogonal Projectors

A Quick Check of the Roadmap — Rear-view Mirror

So far we have reviewed (or quickly introduced) basic linear algebra concepts, *e.g.*

- Vector and Matrix operations, including norms.
- Matrix properties (vocabulary): rank, range, nullspace, domain, Hermitian conjugate (adjoint), unitary...

Then we introduced the idea — from a geometrical perspective — of the Singular Value Decomposition $A = U\Sigma V^*$ of a matrix.

Finally, we connected the SVD and its properties to the majority of the concepts introduced.

In a sense, with the SVD we have extracted all information from the matrix A and we are "done."

A Quick Check of the Roadmap — Looking Forward

Problem#1: We do not have a stable algorithm to compute the SVD. (We don't even know what "stable" means!)

Problem#2: Even when we have such an algorithm (later in the semester), it will turn out to be quite computationally expensive.

The Approach: We will now start building our **computational toolbox** so that in the end we **can** implement a stable, effective algorithm for the SVD.

Along the way we will study other decompositions which may not be as complete as the SVD, but are cheaper to compute and are quite useful in certain applications.

A Quick Check of the Roadmap — The Near Future

- **Projectors**: Orthogonal and non-orthogonal projection matrices.
- **The QR-Factorization**
  - As an idea...
  - Computed using Gram-Schmidt orthogonalization
  - Computed using Householder triangularization
  - Alternative **not** discussed: Computed using Givens rotation ($\approx$ 50% more expensive than Householder, with no additional benefit.)
  - Solving least-squares problems using the QR-factorization

Projections, Projections, Everywhere!!!

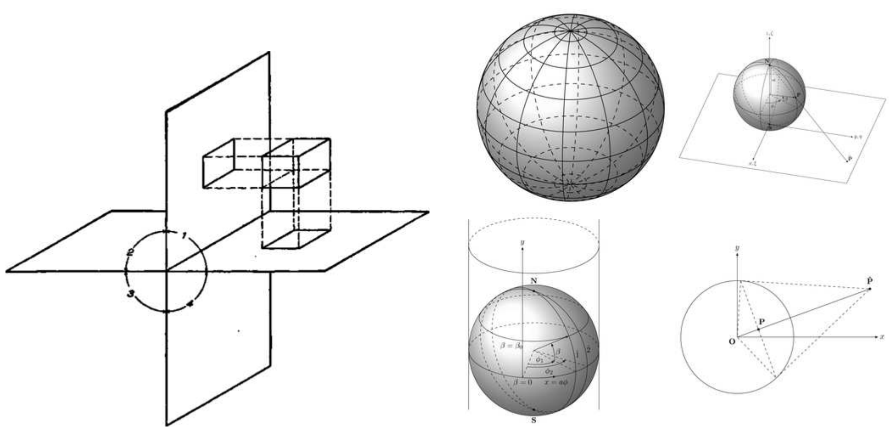


Figure: LEFT— Projecting a geometrical shape onto different planes (the figure itself is a 2D projection of this 3D-to-2D projection!); RIGHT— Map projections; $\mathbb{S}^2 \mapsto \mathbb{R}^2$, and $\mathbb{S}^2 \mapsto \mathbb{R} \times [-\pi, \pi]$.

Projectors — Idempotent Matrices

Definition (Projector)

A **projector** is a square matrix P that satisfies

$$P^2 = P.$$

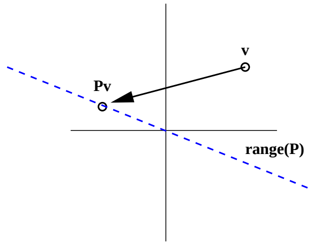
Think, for instance of

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

as the projection of a vector in $\mathbb{R}^3$ onto the x-y plane in $\mathbb{R}^3$: — find a corner in the room; put a broom-stick in the corner and let it point into the room; observe the shadow on the floor. (This is making the assumption that the lighting is laser-based and arranged so that all light-rays go straight from ceiling-to-floor...)

Projectors — Range and Nullspace

More generally, $P : \vec{v} \mapsto P\vec{v}$ maps the vector $\vec{v}$ onto $\mathrm{range}(P)$.



Clearly, once the $\vec{p} = P\vec{v}$ is on $\mathrm{range}(P)$, another projection has no effect, hence

$$P\vec{p} = P^2\vec{v} = P\vec{v} \quad \Leftrightarrow \quad P(P\vec{v} - \vec{v}) = P^2\vec{v} - P\vec{v} = 0$$

Thus $(P\vec{v} - \vec{v}) \in \mathrm{null}(P)$. If we think in terms of the projection being the shadow of a light-source illuminating $\vec{v}$, it means that the direction of the light-rays are described by a vector in $\mathrm{null}(P)$.

Complementary Projectors

If P is a projector, then $(I - P)$ is also a projector

$$(I-P)^2 = I^2 - IP - PI + P^2 = I - 2P + P = I - P$$

$(I - P)$ is the **complementary projector** to P.

We have the following properties

$$\left\{ \begin{array}{rcl} \mathrm{range}(I-P) &=& \mathrm{null}(P) \\ \mathrm{null}(I-P) &=& \mathrm{range}(P) \\ \mathrm{null}(I-P) \cap \mathrm{null}(P) &=& \{\vec{0}\} \\ \mathrm{range}(\mathbf{P}) \cap \mathrm{null}(\mathbf{P}) &=& \{\tilde{\mathbf{0}}\} \end{array} \right.$$

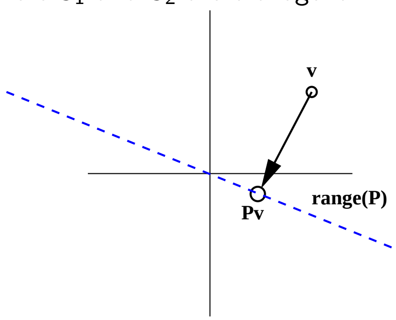
$\mathrm{range}(I-P) \supseteq \mathrm{null}(P)$, since if $P\vec{v} = 0$, then $(I-P)\vec{v} = \vec{v}$
$\mathrm{range}(I-P) \subseteq \mathrm{null}(P)$, since $\forall \vec{v}$, $(I-P)\vec{v} = (\vec{v} - P\vec{v}) \in \mathrm{null}(P)$.

Complementary Projectors — Separation of $\mathbb{C}^m$

! We notice that a projector P separates $\mathbb{C}^m$ into two spaces. !

Conversely, if $S_1, S_2 \subseteq \mathbb{C}^m$ such that $S_1 \cap S_2 = \{\vec{0}\}$, and $S_1 + S_2 = \mathbb{C}^m$, then S_1 and S_2 are **complementary subspaces** and there exists a projector P **onto** S_1 **along** S_2 such that $\mathrm{range}(P) = S_1$, and $\mathrm{null}(P) = S_2$.

An **orthogonal projector** is a projector that projects onto a subspace S_1 along a space S_2, where S_1 and S_2 are orthogonal



Orthogonal Projectors

Warning!!!
Orthogonal projectors are not orthogonal/unitary matrices!!!

An **orthogonal projector** is a projector that is also Hermitian, *i.e.*

$$P^* = P, \quad \text{and} \quad P^2 = P$$

If $P = P^*$, then the inner product of $P\vec{x} \in S_1$ and $(I-P)\vec{y} \in S_2$ is zero:

$$\langle\, P\vec{x}, (I-P)\vec{y}\,\rangle = \vec{x}^* P^*(I-P)\vec{y} = \vec{x}^*(\underbrace{P - P^2}_{P-P})\vec{y} = 0$$

Orthogonal Projectors $\Rightarrow P^* = P$

We now show that if P projects onto S_1 along S_2 ($S_1 \perp S_2$, and S_1 has dimension n), then $P = P^*$ — the construction will give us a very simple characterization of the projector in terms of the SVD!

We construct the SVD of P as follows:

Let $\{\vec{q}_1, \vec{q}_2, \dots, \vec{q}_m\}$ be an orthonormal basis for $\mathbb{C}^m$, where. $\{\vec{q}_1, \vec{q}_2, \dots, \vec{q}_n\}$ is a basis for S_1, and $\{\vec{q}_{n+1}, \vec{q}_{n+2}, \dots, \vec{q}_m\}$ is a basis for S_2. We have

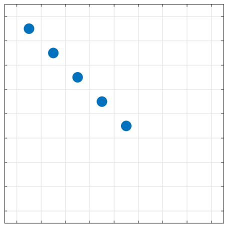
$$\begin{cases} P\vec{q}_j &= \vec{q}_j, \quad j \le n \\ P\vec{q}_j &= 0, \quad j > n \end{cases}$$

Now, let Q be the unitary ($m \times m$) matrix whose jth column is $\vec{q}_j$

Orthogonal Projectors $\Rightarrow P^* = P$

With this construction we have

$$PQ = \begin{bmatrix} | & & | & | & \\ \vec{q}_1 & \dots & \vec{q}_n & \vec{0} & \dots \\ | & & | & | & \end{bmatrix}$$



and, multiplying by Q^* from the left:

$$Q^*PQ = \mathrm{diag}(\underbrace{1, \dots, 1}_{n \text{ ones}}, \underbrace{0, \dots, 0}_{(m-n) \text{ zeros}}) = \Sigma$$

Thus we have constructed an SVD of P:

$$P = Q\Sigma Q^*$$

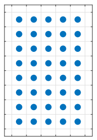
and clearly P is Hermitian

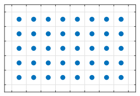
$$P^* = (Q\Sigma Q^*)^* = (Q^*)^*\Sigma^* Q^* = Q\Sigma Q^* = P. \ \square$$

Projection with an Orthonormal Basis

Since some singular values (in Σ) are zero, we can use the reduced SVD instead, *i.e.* we only keep the first n columns in Q, and we end up with

$$P = \widehat{Q}\widehat{Q}^*$$





where the columns of $Q \in \mathbb{C}^{m \times n}$ are orthonormal.

There is nothing magic about orthonormal vectors associated with the SVD — as long as the columns, $\vec{q}_j \in \mathbb{C}^m$, of $\widehat{Q}$ are orthonormal, the matrix $P = QQ^*$ defines an orthogonal projection onto $S_1 = \mathrm{range}(Q)$.

Projection with an Orthonormal Basis — Rank-One Projections

The projection

$$\vec{v} \mapsto P\vec{v} \quad \{\text{defined by}\} \quad QQ^*\vec{v} = \sum_{i=1}^{n} (\vec{q}_i\vec{q}_i^*)\vec{v}$$

can be viewed as a sum on n rank-one projections,

$$P_i = \vec{q}_i\vec{q}_i^*$$

where each such projection isolates the component in a single direction given by $\vec{q}_i$. **These rank-one projectors will show up as building blocks in future algorithms.**

For completeness, we note that the complement of a rank-one projector is a rank-$(m-1)$ projector that eliminates the component in the direction of $\vec{q}_i$

$$P_{\perp\vec{q}_i} = (I - \vec{q}_i\vec{q}_i^*)$$

Projection with a Non-Orthonormal Basis

We can build an orthogonal projector from an arbitrary (not necessarily orthogonal) basis.

Let S_1 be the subspace spanned by the linearly independent vectors $\{\vec{a}_1, \ldots, \vec{a}_n\}$ and let A be the matrix with columns $\vec{a}_j$.

$$\begin{array}{rcl}
\vec{v} & \overset{P}{\mapsto} & \vec{y} \in \mathrm{range}(A), \quad \vec{y} = A\vec{x}, \text{ some } \vec{x} \in \mathbb{C}^n \\
\vec{y} - \vec{v} & \perp & \mathrm{range}(A) \\
& \Leftrightarrow & \vec{a}_j^*(\vec{y} - \vec{v}) = 0, \ \forall j \\
& \Leftrightarrow & \vec{a}_j^*(A\vec{x} - \vec{v}) = 0, \ \forall j \\
& \Leftrightarrow & A^*(A\vec{x} - \vec{v}) = 0 \\
& \Leftrightarrow & A^*A\vec{x} = A^*\vec{v} \\
& \Leftrightarrow & \vec{x} = (A^*A)^{-1}A^*\vec{v} \\
& \Leftrightarrow & \vec{y} = \underbrace{A(A^*A)^{-1}A^*}_{\text{Numerically Dangerous}}\vec{v} = \mathbf{P}\vec{v}
\end{array}$$

Projections: Summary

The key thing we bring from the discussion on projections is the ability to identify how much of the "action" is directed in a certain set of directions, or subspace.

These ideas will be used, explicitly or implicitly, in many algorithms presented in this (and other) classes.

We now turn our attention to on of the "heavy-hitters" among numerical algorithms — the QR-factorization.

The Reduced QR-Factorization

In many application we are interested in the column spaces spanned by a matrix A, *i.e.* the spaces

$$\mathrm{span}(\vec{a}_1) \subseteq \mathrm{span}(\vec{a}_1, \vec{a}_2) \subseteq \mathrm{span}(\vec{a}_1, \vec{a}_2, \vec{a}_3) \subseteq \ldots$$

We may, for instance, be looking for a minimum, or maximum of some quantity over each subspace.

The QR-factorization generates a sequence of orthonormal vectors $\{\vec{q}_1, \vec{q}_2, \vec{q}_3, \ldots\}$ that spans these spaces, *i.e.*

$$\mathrm{span}(\vec{q}_1, \vec{q}_2, \ldots, \vec{q}_k) = \mathrm{span}(\vec{a}_1, \vec{a}_2, \ldots, \vec{a}_k), \quad k = 1, \ldots, n$$

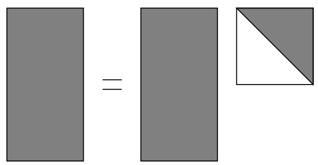
The reason for doing this is that it is much easier to work in an orthonormal basis.

The Reduced QR-Factorization — The Idea

$$\begin{array}{rcl}
\mathrm{span}(\vec{q}_1) = \mathrm{span}(\vec{a}_1) & \Rightarrow & \vec{a}_1 = r_{11}\vec{q}_1 \\
\mathrm{span}(\vec{q}_1, \vec{q}_2) = \mathrm{span}(\vec{a}_1, \vec{a}_2) & \Rightarrow & \vec{a}_2 = r_{12}\vec{q}_1 + r_{22}\vec{q}_2 \\
\mathrm{span}(\vec{q}_1, \ldots, \vec{q}_3) = \mathrm{span}(\vec{a}_1, \ldots, \vec{a}_3) & \Rightarrow & \vec{a}_3 = r_{13}\vec{q}_1 + r_{23}\vec{q}_2 + r_{33}\vec{q}_3 \\
& \vdots & \\
\mathrm{span}(\vec{q}_1, \ldots, \vec{q}_n) = \mathrm{span}(\vec{a}_1, \ldots, \vec{a}_n) & \Rightarrow & \vec{a}_n = r_{1n}\vec{q}_1 + \cdots + r_{nn}\vec{q}_n
\end{array}$$

In matrix notation, with $A \in \mathbb{C}^{m\times n}$, $\widehat{Q} \in \mathbb{C}^{m\times n}$ with orthonormal columns, $\widehat{R} \in \mathbb{C}^{n\times n}$

$$\mathbf{A} = \widehat{\mathbf{Q}}\widehat{\mathbf{R}}$$



The Full QR-Factorization

As for the SVD, we can extend the QR-factorization by padding $\widehat{Q}$ with an additional $(m-n)$ orthonormal columns, and zero-padding $\widehat{R}$ with an additional $(m-n)$ rows of zeros:

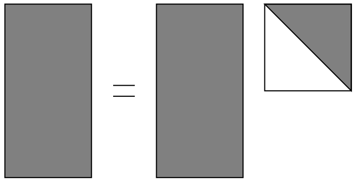


Figure: The Reduced QR-Factorization, $A = \widehat{Q}\widehat{R}$

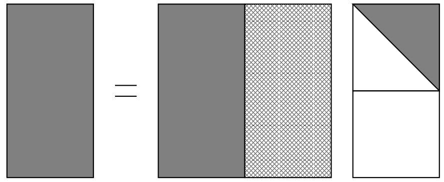


Figure: The Full QR-Factorization, $A = QR$

In the full QR-factorization, the columns $\vec{q}_j$, $j > n$ are orthogonal to $\mathrm{range}(A)$. If $\mathrm{rank}(A) = n$, they are an orthonormal basis for $\mathrm{range}(A)^{\perp} = \mathrm{null}(A^*)$, the space orthogonal to $\mathrm{range}(A)$.

Building the QR-Factorization — Gram-Schmidt Orthogonalization

The equations on slide 20 outline a method for **computing** reduced QR-factorizations.

At the kth step, we are looking to construct $\vec{q}_k \in \mathrm{span}(\vec{a}_1, \ldots \vec{a}_k)$ such that $\vec{q}_k \perp \mathrm{span}(\vec{q}_1, \ldots \vec{q}_{k-1})$

We simply take $\vec{a}_k$, and subtract all the projections onto the directions $\vec{q}_1, \ldots \vec{q}_{k-1}$, and then normalize the resulting vector

$$(*) \quad \begin{aligned} \vec{v}_k &= \vec{a}_k - (\vec{q}_1\vec{q}_1^*)\vec{a}_k - \cdots - (\vec{q}_{k-1}\vec{q}_{k-1}^*)\vec{a}_k \\ \vec{q}_k &= \vec{v}_k / \|\vec{v}_k\|_2 \end{aligned}$$

Computationally, it is more efficient to compute

$$(*') \quad \vec{v}_k = \vec{a}_k - \vec{q}_1(\vec{q}_1^*\vec{a}_k) - \cdots - \vec{q}_{k-1}(\vec{q}_{k-1}^*\vec{a}_k)$$

`Algorithm:` Classical Gram-Schmidt ∃ Movie

We summarize our findings:

Algorithm (Classical Gram-Schmidt)

for $k \in \{1, \ldots, n\}$ **do**
  $\vec{v}_k \leftarrow \vec{a}_k$
  for $i \in \{1, \ldots, k-1\}$ **do**
    $r_{ik} \leftarrow \vec{q}_i^*\vec{a}_k$
    $\vec{v}_k \leftarrow \vec{v}_k - r_{ik}\vec{q}_i$
  end for
  $r_{kk} \leftarrow \|\vec{v}_k\|_2$
  $\vec{q}_k \leftarrow \vec{v}_k / r_{kk}$
end for

Mathematically, we are done. Numerically, however, we can run into trouble due to roundoff errors.

The QR-Factorization: Existence and Uniqueness 1 of 2

Theorem (Existence of the QR-Factorization)

Every $A \in \mathbb{C}^{m \times n}$ $(m \geq n)$ has a full QR-factorization, hence also a reduced QR-factorization.

Theorem (Uniqueness of the QR-Factorization)

Every $A \in \mathbb{C}^{m \times n}$ $(m \geq n)$ of full rank has a unique reduced QR-factorization $A = \widehat{Q}\widehat{R}$, with $r_{kk} > 0$.

The QR-Factorization: Existence and Uniqueness — 2 of 2

1. If A is full rank, the Gram-Schmidt algorithm gives the unique reduced QR-factorization.
2. If A does not have full rank, then $\vec{v}_k = 0$ can occur during the iteration; if it does set $\vec{q}_k$ to be an arbitrary vector* orthogonal to $\mathrm{span}\,(\vec{q}_1, \ldots \vec{q}_{k-1})$, and proceed.
3. **For Full QR factorization**, when $m > n$, follow Gram-Schmidt as described until $j = n$, then take an addition $(m - n)$ steps, introducing arbitrary orthogonal $\vec{q}_k$ in each step.

* Column pivoting (exchanges) may be necessary.

Solving $A\vec{x} = \vec{b}$ by QR-Factorization

If we have a QR-factorization algorithm handy, then we have the following "algorithm" for solving $A\vec{x} = \vec{b}$

1. Compute the QR-factorization $A = QR$.
2. Compute $\vec{y} = Q^*\vec{b}$.
3. Solve $R\vec{x} = \vec{y}$ for $\vec{x}$.

Note: Computing $Q^*\vec{b}$ is just a multiplication with a unitary matrix. Since $|\det(Q^*)| = 1$ this completely numerically stable in the sense that errors will not be magnified. (We will quantify this soon.)

Note: Solving $R\vec{x} = \vec{y}$ is very easy (backward substitution) since R is upper triangular.

Note: The bulk of the work is in computing the QR-factorization (2–3 times that of Gaussian Elimination).

Homework #3 — Due Date in Canvas/Gradescope

Implement the reduced QR-factorization by **Classical Gram-Schmidt**.

1. Write a function which given $A \in \mathbb{C}^{m\times n}$ computes $Q \in \mathbb{C}^{m\times n}$, and $R \in \mathbb{C}^{n\times n}$ — in matlab/python you want to start something like this (*e.g.* file: `qr_cgs.m`, or `qr_cgs.py`):

```
function [Q,R] = qr_cgs(A)
% Indentation does not matter...
% Implicit Return of Results
```

```
def qr_cgs(A)
    # Indentation matters
    # Explicit Return of Results
    return Q, R
```

See `help function` in matab, or `python_functions` (clickable) for help on writing functions.

2. Validate your function — test that (i) $(A - QR) \approx 0$; (ii) Q is unitary; and (iii) R upper triangular. Show 3 test cases for (3×3), (5×5), and (251×251) matrices.

3. Compare the result for the (3×3), (5×5) cases with the built-in ("library") version of the QR-factorization; comment on the similarities/differences.
See `help qr in matlab, or numpy.linalg.qr` (clickable).

4. Can you find a non-zero matrix where your QR-factorization breaks?

∞. Hand in your code, and your validation/test-cases.

∞^∞. Appropriately "tag" all pages with the corresponding question(s) in Gradescope.

Homework AI-Policy Spring 2024

AI-era Policies — SPRING 2024

AI-3 Documented: *Students can use AI in any manner for this assessment or deliverable, but they must provide appropriate documentation for all AI use.*

This applies to ALL MATH-543 WORK during the SPRING 2024 semester.

The goal is to leverage existing tools and resources to generate HIGH QUALITY SOLUTIONS to all assessments.

You MUST document what tools you use and HOW they were used (including prompts); AND how results were VALIDATED.

BE PREPARED to DISCUSS homework solutions and AI-strategies. **Participation in the in-class discussions will be an essential component of the grade for each assessment.**