Numerical Matrix Analysis
Lecture Notes #16 — Review:
SVD, QR, Least Squares, Conditioning and Stability

Peter Blomgren,
⟨blomgren.peter@gmail.com⟩

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 2016
Outline

1 The Big Picture
   - Model Problem and Attacks; Analysis
   - Building Blocks; Tools

2 Fundamentals
   - Basic Linear Algebra
   - The SVD; Projections
   - QR-Factorization

3 Fundamentals, ctd.
   - Linear Least Squares
   - Conditioning, Stability and Accuracy
   - Error Analysis & Stability
The Linear Least Squares Problem

\[
\min_{\bar{x} \in \mathbb{C}^n} \| A\bar{x} - \bar{b} \|_2, \quad A \in \mathbb{C}^{m \times n}, \quad m \geq n, \quad \text{rank}(A) = n
\]

Attacks

<table>
<thead>
<tr>
<th>Normal Equations</th>
<th>QR-Factorization</th>
<th>The SVD</th>
</tr>
</thead>
</table>

Methods

<table>
<thead>
<tr>
<th>Brute Force</th>
<th>Gram-Schmidt Orthogonalization</th>
<th>Householder Triangularization</th>
<th>“magic”</th>
</tr>
</thead>
</table>

Modes

<table>
<thead>
<tr>
<th>Explicit $Q$</th>
<th>Implicit $Q^*\bar{b}$</th>
</tr>
</thead>
</table>
The Big Picture: Analysis

Conditioning

The inherent difficulty of the mathematical problem
Sensitivity to perturbations; quantified by the condition number, $\kappa$

Stability

The robustness of the algorithm

Backward Stability

$$\tilde{f}(\bar{x}) = f(\bar{x}), \quad \frac{||\tilde{x} - \bar{x}||}{||\bar{x}||} = O(\epsilon_{mach})$$

Stability

$$\frac{||\tilde{f}(\bar{x}) - f(\bar{x})||}{||f(\bar{x})||}, \quad \frac{||\bar{x} - \bar{x}||}{||\bar{x}||} = O(\epsilon_{mach})$$

Accuracy

For a backward stable algorithm, the accuracy is

$$\frac{||\tilde{f}(\bar{x}) - f(\bar{x})||}{||f(\bar{x})||} = O(\kappa(\bar{x})\epsilon_{mach})$$
The SVD

Here used primarily (so far) for matrix understanding, expression of the condition number of a matrix, simplification of proofs; “every matrix is diagonal.”

Projectors

$P^2 = P$. Orthogonal if $P^* = P$. Can be formed using an orthogonal ($P = QQ^*$), or non-orthogonal ($P = A(A^*A)^{-1}A^*$) basis.

Floating Point

A source of **unavoidable** errors in representation of numerical values, and computations.

Norms

Matrix and vector norms give us the fundamental measurements of size and distance in our vector spaces.
Basic Linear Algebra, etc.

- Fundamental matrix/vector operations, orthogonality, orthonormality, inner products, the angle between two vectors, Hermitian transpose, linear independence, basis for a space, unitary matrices.

- Vector and matrix norms, especially the $\| \cdot \|_1$, $\| \cdot \|_2$, and $\| \cdot \|_\infty$ norms, also the Frobenius norm of a matrix; 2-norm invariance under multiplication by a unitary matrix.

- Vector- and matrix-norm inequalities; Cauchy-Bunyakovskyy-Schwarz.

- The SVD as a tool for simplifying analysis and understanding of matrix properties. Geometric understanding.

Unlikely to show up as explicit questions on the midterm.
The SVD $A = U \Sigma V^*$

- (For now) a theoretical tool.
- The full and reduced SVD.
- Expressing range$(A)$ and null$(A)$ in terms of the components of the SVD.
- The singular values $\sim$ rank$(A)$, $\kappa(A)$, $\|A\|_2$, and $\|A\|_F$. 

Peter Blomgren, ⟨blomgren.peter@gmail.com⟩
Definition (Projector)

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

$$P^2 = P$$

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;

$$\iff P = P^*$$
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 \textbf{complementary projector} to \( P \).

We have the following properties

\[
\begin{align*}
\text{range}(I - P) &= \text{null}(P) \\
\text{null}(I - P) &= \text{range}(P) \\
\text{null}(I - P) \cap \text{null}(P) &= \{0\} \\
\text{range}(P) \cap \text{null}(P) &= \{0\}
\end{align*}
\]

\[
\text{range}(I - P) \supseteq \text{null}(P), \text{ since if } P\vec{v} = 0, \text{ then } (I - P)\vec{v} = \vec{v}
\]

\[
\text{range}(I - P) \subseteq \text{null}(P), \text{ since } \forall \vec{v}, (I - P)\vec{v} = \vec{v} - P\vec{v} \in \text{null}(P)
\]
The projection

$$\vec{v} \rightarrow P\vec{v} = QQ^*\vec{v} = \sum_{i=1}^{n}(\bar{q}_i\bar{q}_i^*)\vec{v}$$

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

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

where each such projection isolates the component in a single direction of $\bar{q}_i$. 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 $\bar{q}_i$

$$P_{\perp\bar{q}_i} = I - \bar{q}_i\bar{q}_i^*$$
The idea; what does the factorization look like? How can it be used to solve $A\tilde{x} = \tilde{b}$? (When $A$ is square? Rectangular?) Why not “just” compute the SVD and be done?

Building blocks: projectors; what choice is made? Why?

Projectors; orthogonal — properties? How to build one from an orthonormal, and general basis?

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.
The Reduced QR-Factorization

\[
\langle \bar{q}_1 \rangle = \langle \bar{a}_1 \rangle \quad \Rightarrow \quad \bar{a}_1 = r_{11} \bar{q}_1 \\
\langle \bar{q}_1, \bar{q}_2 \rangle = \langle \bar{a}_1, \bar{a}_2 \rangle \quad \Rightarrow \quad \bar{a}_2 = r_{12} \bar{q}_1 + r_{22} \bar{q}_2 \\
\langle \bar{q}_1, \ldots, \bar{q}_3 \rangle = \langle \bar{a}_1, \ldots, \bar{a}_3 \rangle \quad \Rightarrow \quad \bar{a}_3 = r_{13} \bar{q}_1 + r_{23} \bar{q}_2 + r_{33} \bar{q}_3 \\
\vdots
\]

\[
\langle \bar{q}_1, \ldots, \bar{q}_n \rangle = \langle \bar{a}_1, \ldots, \bar{a}_n \rangle \quad \Rightarrow \quad \bar{a}_n = r_{1n} \bar{q}_1 + \cdots + r_{nn} \bar{q}_n
\]

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

\[
A = \hat{Q} \hat{R}
\]
As for the SVD, we can extend the QR-factorization by “fleshing out” $\hat{Q}$ with an additional $(m - n)$ orthonormal columns, and zero-padding $\hat{R}$ with an additional $(m - n)$ rows of zeros:

$$
\begin{array}{cccc}
\vdots & \vdots & \vdots & \vdots \\
\vdots & \vdots & \vdots & \vdots \\
\vdots & \vdots & \vdots & \vdots \\
\vdots & \vdots & \vdots & \vdots \\
\vdots & \vdots & \vdots & \vdots \\
\vdots & \vdots & \vdots & \vdots \\
\vdots & \vdots & \vdots & \vdots \\
\vdots & \vdots & \vdots & \vdots \\
\end{array}
\quad = 
\begin{array}{cccc}
\vdots & \vdots & \vdots & \vdots \\
\vdots & \vdots & \vdots & \vdots \\
\vdots & \vdots & \vdots & \vdots \\
\vdots & \vdots & \vdots & \vdots \\
\vdots & \vdots & \vdots & \vdots \\
\vdots & \vdots & \vdots & \vdots \\
\vdots & \vdots & \vdots & \vdots \\
\vdots & \vdots & \vdots & \vdots \\
\end{array}
\quad = 
\begin{array}{cccc}
\vdots & \vdots & \vdots & \vdots \\
\vdots & \vdots & \vdots & \vdots \\
\vdots & \vdots & \vdots & \vdots \\
\vdots & \vdots & \vdots & \vdots \\
\vdots & \vdots & \vdots & \vdots \\
\vdots & \vdots & \vdots & \vdots \\
\vdots & \vdots & \vdots & \vdots \\
\vdots & \vdots & \vdots & \vdots \\
\end{array}
$$

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

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

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

- Classical Gram-Schmidt
  - Think of how we use projectors to build CGS.
  - Problem: Loss of orthogonality in $Q$,
  - Problem: Errors in $R \sim \mathcal{O}(\sqrt{\epsilon_{\text{mach}}})$.

**Figure:** $Q^*Q \neq I$, illustrating the loss of orthogonality in CGS.

**Figure:** The blue circles illustrate that we only reach an accuracy level of $\sim \sqrt{\epsilon_{\text{mach}}}$ for CGS.

Peter Blomgren, {blomgren.peter@gmail.com}
Algorithms for the QR-Factorization

- **Modified Gram-Schmidt**
  - Mathematically equivalent to CGS; only a slight re-ordering of operations,
  - Numerically stable — If used in “implicit mode” for least squares problems.

**Figure:** $Q^*Q \approx I$, illustrating the improved orthogonality properties in MGS.

**Figure:** The red crosses illustrate the that we reach an accuracy level of $\sim \epsilon_{\text{mach}}$ for MGS.
Algorithms for the QR-Factorization

- **Householder Triangularization**
  - Use of reflectors, closely related to projectors. Non-uniqueness of reflectors: understand the choice of reflector!
  - The basic implementation is $Q$-less! Understand how to get the action $Q^*\bar{b}$ "implicitly," and how to add the explicit formation of $Q$ to the algorithm.
  - Numerically stable, and almost perfect orthogonality.
Least Squares Problems

- Understand the formulation and interpretation of the least squares problems — \( \text{range}(A), \min \| \cdot \|_2, A\tilde{x} = \bar{P}\bar{y} \), the residual and orthogonality...

- Know how to set up a least squares problem for fitting a low degree polynomial to a given data set.

- What is the solution of the least squares problem, as expressed in terms of the results of the QR-factorization, Singular Value Decomposition, and the Normal Equations?

- Rough work comparison for the different solutions:

\[
\text{QR} \sim 2(\text{NE}), \quad \text{SVD} \leq 10(\text{QR}), \quad \text{but } \text{SVD} \approx \text{QR when } m \gg n
\]
Cornerstones: Conditioning, Stability and Accuracy

- Absolute and relative condition numbers; definitions; use of the Jacobian when $f : X \rightarrow Y$ is differentiable.
- Note that the condition number may be a function of $\bar{x}$, i.e. a problem may be well-conditioned for some range of inputs, but ill-conditioned for other inputs.
- **Building block** — The condition number of a matrix: in terms of $\|A\|$, $\|A^{-1}\|$ ($\|A^\dagger\|$ as appropriate), and $\sigma*$.
- The Floating Point Axioms.
- Absolute and relative errors.
- Accuracy of an algorithm.
- Stability and Backward Stability.
Conditioning of a Problem

- The absolute and relative condition numbers (what kind is most useful in the context of computational science, why?) Definitions, and ability to compute for simple problems.

- Differentiability and non-differentiability (impact)

- What is a “small” condition number? A large one?

- The condition number is a measure of the inherent difficulty of the (mathematical) problem.

- Conditioning of basic linear algebra operations, and the condition number of a matrix.
Floating point arithmetic

- The axioms, and impact on computations.

Stability

- Stability is a statement about the quality of an algorithm.
- The formal and informal definitions of stability and backward stability.

- “A **stable algorithm gives approximately the right answer, to approximately the right question.**”

- “A **backward stable algorithm gives exactly the right answer, to approximately the right question.**”
• *Simple* demonstrations of backwards stability. — Backward error analysis.

Accuracy

• The connection between conditioning, stability, and accuracy.

• Forward and backward errors; impact on (indication of) stability.

• The use of backward stable linear algebra building blocks to solve $A\bar{x} = \bar{b}$, etc...
Backward Error Analysis ⇝ Backward Stability

- Understand how it is done: Interpretation of floating point errors as perturbations.
- For $\tilde{f} : X \to Y$, if $\dim(Y) > \dim(X)$, then $\tilde{f}$ is rarely backward stable.
- Accuracy(Stability, Conditioning).
- Backward and forward errors; dependence on the condition number; impact on (backward) stability.
Conditioning of Least Squares Problems

- Conditioning of Least Squares problems; best and worst case scenarios.
- Impact of data sampling (measurements) and model selection...
- Unstable solution strategies; — Normal Equations.
- Backward stable solution strategies; computational effort, solution quality; expected accuracy.
- Compare with conditioning for the normal equations.
- Notice how modified Gram-Schmidt must be used in “implicit mode” in order to provide a backward stable method for solving least squares problems.