Least Squares Approach for tomographic reconstruction

Very simplified test problem illustrating the use of the SVD factorization in analyzing a linear system.

Consider an object made up of square blocks in a $3 \times 3$ arrangement, with unknown densities in each block:

\[
\begin{array}{ccc}
  x_1 & x_2 & x_3 \\
  x_4 & x_5 & x_6 \\
  x_7 & x_8 & x_9 \\
\end{array}
\]

Recording row sums $r_1, r_2, r_3$ and column sums $s_1, s_2, s_3$, can we from these recover $x_1, \ldots, x_9$? Is the problem solvable for all values of $r_1, r_2, r_3, s_1, s_2, s_3$? If not, what are the conditions that make the system solvable?

If it is solvable, how many free parameters are there in the solution space?

Without a systematic approach (which SVD provides), the questions are hard to answer accurately.

Some scattered observations:

1. We can write the system in standard form as

\[
\begin{bmatrix}
  1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
  0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
  0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\
  0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
  0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
\end{bmatrix}
\begin{bmatrix}
  x_1 \\
  x_2 \\
  x_3 \\
  x_4 \\
  x_5 \\
  x_6 \\
  x_7 \\
  x_8 \\
  x_9 \\
\end{bmatrix}
= \begin{bmatrix}
  r_1 \\
  r_2 \\
  r_3 \\
  s_1 \\
  s_2 \\
  s_3 \\
\end{bmatrix}
\]

or $Ax = b$
2. Adding the first three equations give
\[ x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 + x_9 = r_1 + r_2 + r_3 \] (2)
and adding the next three give
\[ x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 + x_9 = s_1 + s_2 + s_3 \] (3).
The left-hand sides of (2) and (3) are the same, so we clearly have a conflict if
\[ r_1 + r_2 + r_3 \neq s_1 + s_2 + s_3 \] (5).
The system (1) does not always have a solution.
Is (5) the only constraint?

3. Say that we obey \( r_1 + r_2 + r_3 = s_1 + s_2 + s_3 \), more specifically, have \( r_1 = r_2 = r_3 = 1 \) and \( s_1 = s_2 = s_3 = 1 \). Then the following 6 very different objects are all valid:
\[
\begin{bmatrix}
1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}, \begin{bmatrix}
0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}, \begin{bmatrix}
0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}, \begin{bmatrix}
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0
\end{bmatrix}, \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0
\end{bmatrix}, \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0
\end{bmatrix}.
\]

Does this mean that there are 6 free parameters in the solution space? It will turn out that we have a 4-parameter space, clearly we need a systematic approach.

These questions (and more) become answered if we consider the SVD factorization
\[ U S V^T = A \]
and remember how to interpret the information contained in the \( U, S, V^T \)-matrices.
There are the 'rules' we will apply.

SINGULAR VALUE DECOMPOSITION (SVD)

An illustration of the bases of the four fundamental subspaces

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

- basis for row-space of $A$
- basis for null-space of $A$
- first $r$ diagonal elements = singular values of $A$; all other entries zero
What are the constraints the RHS needs to satisfy so there are solutions?

The last column of \( U \) tells that \( 0.4802 \cdot \begin{bmatrix} 1 \\ -1 \\ -1 \end{bmatrix} \) (when transposed) is the left null vector.

If we consider \( Ax = b \) and multiply by \( \begin{bmatrix} 1 \\ 1 \\ 1 \\ -1 \\ -1 \end{bmatrix} \) from the left, we get

\[
\begin{bmatrix} 1 \\ 1 \\ 1 \\ -1 \\ -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ -1 \\ -1 \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix}
\]

\[= \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \] since left null vector.

So, \( 0 = \begin{bmatrix} 1 \\ 1 \\ 1 \\ -1 \\ -1 \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix} \) and we have obtained the same constraint:

\[r_1 + r_2 + r_3 = s_1 - s_2 - s_3 = 0.\]

But now we also know \( \begin{bmatrix} 1 \\ 1 \\ 1 \\ -1 \\ -1 \end{bmatrix} \) was the only left null vector - so there are no further constraints on the RHS.

If the constraint is satisfied, how can we pick up solutions?

We have \( Ax = b \rightarrow \)

\[
\begin{bmatrix} u_1 & u_2 & u_3 & u_4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix}
\]

\[= \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \] call this (y)

Multiply by \( U^\top \) from left, \( \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} \) call \( \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix} = [E] \).
The system now is:

\[
\begin{bmatrix}
\vdots & \\
0 & \\
\vdots & \\
\end{bmatrix}
\begin{bmatrix}
y_1 \\
y_2 \\
\vdots \\
\end{bmatrix} = 
\begin{bmatrix}
\ell \\
\ell \\
\end{bmatrix}.
\]

System is assumed solvable, so the last component of \( \ell \) is zero.

It is clear that the first 5 components of \( \ell \) are determined, but the 4 remaining are free - we can give them any values and the system is still satisfied.

Hence, there is a 4-parameter space of solutions.

Looking back at the 6 matrices on page 2, (call them \( P_1, P_2, \ldots, P_6 \)), they can't have been linearly independent.

Indeed, we can note:

\[
P_1 - P_2 - P_3 + P_4 + P_5 - P_6 = \begin{bmatrix}
\cdots \\
\cdots \\
\cdots \\
\cdots \\
\cdots \\
\cdots \\
\cdots \\
\cdots \\
\end{bmatrix}.
\]

Still, SVD gave by far the most clear answers.

We could have seen simpler still that there was a 41-dimensional solution space. The last 41 columns of \( V \) (rows of \( V^T \)) are nullvectors:

\[
\begin{bmatrix}
A \\
\vdots \\
\end{bmatrix}
\begin{bmatrix}
\vdots \\
\vdots \\
\vdots \\
\vdots \\
\vdots \\
\cdots \\
\cdots \\
\end{bmatrix} = 
\begin{bmatrix}
\cdots \\
\cdots \\
\cdots \\
\cdots \\
\cdots \\
\cdots \\
\cdots \\
\end{bmatrix}.
\]

So any multiples of these columns can be added to a solution and we have again a solution.
Suppose there is a conflict, but we want a least square solution — how can we obtain that?

This situation arises most often when there are more equations than unknowns — so we write the system \( [A] [x] = [b] \).

Solving this with SVD works fine, but is "over-kill" — the simpler \( A = QR \) does the same job easier.

**Lemma:** The 2-norm \( \|x\|_2 = \sqrt{x_1^2 + x_2^2 + \ldots + x_n^2} \) of a vector does not change if we multiply \( x \) by a unitary matrix \( U \), i.e.

\[
\|Ux\|_2 = \|x\|_2
\]

**Proof of Lemma:**

We can write

\[
\|x\|_2^2 = x_1^2 + x_2^2 + \ldots + x_n^2 = x^T x
\]

Therefore

\[
\|Ux\|_2^2 = (Ux)^T(Ux) = x^T U^T U x = x^T x = \|x\|_2^2
\]

The least square solution to \( [A] [x] = [b] \) is the vector \( x \) such that

\[
(y) = (A)[x] - [b]
\]

has \( \|y\|_2 \) as small as possible.

Let \( (A) = [Q] [R] \).
We don't change $\|\cdot\|_2$ if we multiply by $Q^T$ from left (due to the Lemma). So:

$$Q^T y = \begin{pmatrix} R' \mid x \end{pmatrix} - \begin{pmatrix} 0^T \mid b \end{pmatrix} \quad \text{call this } \begin{pmatrix} u \end{pmatrix}.$$ 

We clearly make the $\|\cdot\|_2$ of $R x - v$ as small as possible if we choose $x$ to cancel out all the top entries of $v$; the bottom ones can't be done anything about by varying $x$.

The 'classical' least square solution method for $A x = b$ is to form and solve the 'normal equations' $\begin{pmatrix} A^T A \end{pmatrix} x = \begin{pmatrix} A^T b \end{pmatrix}$

(which is a square system).
This however can be numerically unstable—OK by hand, working exactly, but unsuitable on a computer for large problems.

**Question:** Is there an easy way to show that this normal equation approach indeed solves the least squares problem?

We show next it leads to the same linear system as the QR approach, so it must be mathematically equivalent.
The normal equations were
\[ A^T A \mathbf{x} = A^T \mathbf{b} \]
Write here \( A = QR \) the system above then becomes
\[
\begin{bmatrix}
\mathbf{Q}^T \\
\mathbf{R}^T
\end{bmatrix}
\begin{bmatrix}
\mathbf{Q} \\
\mathbf{R}
\end{bmatrix}
\begin{bmatrix}
\mathbf{x}^T \\
\mathbf{b}^T
\end{bmatrix}
= 
\begin{bmatrix}
\mathbf{Q}^T \\
\mathbf{R}^T
\end{bmatrix}
\begin{bmatrix}
\mathbf{Q}^T \\
\mathbf{R}^T
\end{bmatrix}
\begin{bmatrix}
\mathbf{Q} \\
\mathbf{R}
\end{bmatrix}
\begin{bmatrix}
\mathbf{x}^T \\
\mathbf{b}^T
\end{bmatrix}
\]
This simplifies to (also setting rid of zero blocks in \( \mathbf{R} \) \( \mathbf{R}^T \));
\[
\begin{bmatrix}
\mathbf{R}^T \\
\mathbf{R}
\end{bmatrix}
\begin{bmatrix}
\mathbf{x}^T \\
\mathbf{b}^T
\end{bmatrix}
= 
\begin{bmatrix}
\mathbf{R}^T \\
\mathbf{R}^T
\end{bmatrix}
\begin{bmatrix}
\mathbf{Q}^T \\
\mathbf{Q}^T
\end{bmatrix}
\begin{bmatrix}
\mathbf{x}^T \\
\mathbf{b}^T
\end{bmatrix}
\]
The \( \begin{bmatrix}
\mathbf{R}^T \\
\mathbf{R}
\end{bmatrix} \) matrix can be removed from both sides, and we are left with exactly the same equations for \( \mathbf{x} \) as we obtained with the QR method. So the normal equations must also provide the problem's least squares solution.