More Linear Algebra

Singular Value Decomposition (SVD)

“The highpoint of linear algebra” – Gilbert Strang

Any \( m \times n \) matrix \( A \) can be decomposed into:

\[
A = U \Sigma V^T
\]

- \( U : m \times m \) : columns are left singular vectors
- \( \Sigma : m \times n \) : diagonal : singular values
- \( V : n \times n \) : columns are right singular vectors

\( \sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r > 0, r = \text{rank}(A) \)

Economy version \( A = U_r \Sigma_r V_r^T \)

\( U, V \) orthogonal : \( U^T U = I_{m \times m}, V^T V = I_{n \times n} \)

Column Space: look at \( Ax \)

\[
Ax = U \Sigma V^T x, \quad \text{and let } y = V^T x
\]

\[
= \begin{bmatrix}
\sigma_1 u_1 & \ldots & \sigma_r u_r & 0 & \ldots & 0 \\
0 & \ldots & \ldots & \ldots & \ldots & 0 \\
0 & \ldots & \ldots & \ldots & \ldots & 0
\end{bmatrix} y
\]

so \( \text{Col}(A) = \text{Col}(U_r) \). In fact, \( u_1, \ldots, u_r \) form an orthonormal basis for \( \text{Col}(A) \).

Nullspace: look at

\[
Ax = 0
\]

\[
\Rightarrow U_r \Sigma_r V_r^T x = 0
\]
pre-multiply by $U_i^\top$: $\Sigma_r V_r^\top x = 0$
pre-multiply by $\Sigma_r^{-1}$: $V_i^\top x = 0$
i.e. want $x$ to be orthogonal to $v_1, \ldots, v_r$
That’s precisely $v_{r+1}, \ldots, v_n$, since $V$ is orthogonal!
Thus, $v_{r+1}, \ldots, v_n$ form an orthonormal basis for $\text{Null}(A)$.

Consider
$$A^\top A = \left(U \Sigma V^\top\right)^\top \left(U \Sigma V^\top\right) = V \Sigma^2 U^\top U \Sigma V^\top = V \Sigma^2 V^\top$$
But this is the eigen-decomposition of $A^\top A$! So $V$ is the eigenvector matrix of $A^\top A$
$\Sigma^2$ is the eigenvalue matrix of $A^\top A$ i.e. singular values are positive square roots of eigenvalues.

Similarly, consider
$$AA^\top = U \Sigma V^\top V \Sigma^\top U^\top = U \Sigma^2 U^\top$$
So $U$ is the eigenvector matrix for $AA^\top$ with same eigenvalues.
In general, for $m \times n$ $A$:
$$Ax = U \Sigma V^\top x$$
$$\quad = (\text{rotate in } \mathbb{R}^m) (\text{scale}) (\text{rotate in } \mathbb{R}^n) x$$

**Low-rank approximation**

SVD provides the best lower-rank approximation to $A$, i.e. rank $k$ approx. $A_k = U_k \Sigma_k V_k^\top$.
The idea is to use only the first $k$ singular values/vectors, so that $A_k \approx A$.

Instead of storing $A$ : $mn$ numbers
store $u_1, \ldots, u_k$ : $mk$ numbers
Use SVD for compression: $+ \sigma_1, \ldots, \sigma_k$ : $k$ numbers
$+ v_1, \ldots, v_k$ : $nk$ numbers
$= (m + n + 1)k$ numbers

**Use SVD to filter noise**

Typically, small singular values are caused by noise.
using rank $k$ approx ($k < r$), removes noise.

**Linear Equations Revisited: $Ax = b$**

Key: solution only when $b \in \text{Col}(A)$

Case 1. $A$ $n \times n$ and invertible. Then unique solution : $x = A^{-1}b$
$\text{rank}(A) = n, \text{Col}(A) = \mathbb{R}^n$
Case 2. A \( n \times n \) and singular. \( \text{rank}(A) = r < n \), \( \text{nullity} = n - r \)

Two possibilities:

(a) \( b \in \text{Col}(A) \) : many solutions.

(b) \( b \notin \text{Col}(A) \) : no exact solution, closest solution.

(a) \( b \in \text{Col}(A) \) : SVD gives particular solution \( x_p \) such that \( Ax_p = b \)

But we can add any vector from Nullspace, \( x_n \), since

\[
A(x_p + x_n) = Ax_p + Ax_n = b + 0
\]

\( \therefore \) Infinitely many solutions!

What is the SVD solution? Invert only in rank \( r \) subspace

\( A = U\Sigma V^\top \) (all \( n \times n \))

where \( \Sigma = \begin{bmatrix} \sigma_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_r \end{bmatrix} \)

Let \( A^\dagger = V\Sigma^\dagger U^\top \), where \( \Sigma^\dagger = \begin{bmatrix} \frac{1}{\sigma_1} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \frac{1}{\sigma_r} \end{bmatrix} \)

Then \( x_p = A^\dagger b \). \( A^\dagger \) : pseudoinverse. See Figure 1.

(b) \( b \notin \text{Col}(A) \) : No exact solution, but can find \( b' \in \text{Col}(A) \) closest to \( b \)

Solution \( x' = A^\dagger b = V\Sigma U^\top b \)

Case 3. A \( m \times n \) with \( m < n \) “underconstrained” fewer equations than unknowns.

\( r = \text{rank}(A) \leq \min(m,n) \), i.e. \( r < n \), so Nullspace is not trivial. \( \text{Col}(A) \subseteq \mathbb{R}^m \)

Situation similar to the previous case, either \( b \in \text{Col}(A) \) or \( b \notin \text{col}(A) \)

In practice, usually \( r = m \), so that \( b \in \text{Col}(A) \), i.e. many solutions

Case 4. A \( m \times n \) with \( m > n \) “overconstrained”, more equations than unknowns. rank, \( r \),

is at most, \( n \). Therefore, \( \text{Col}(A) \subset \mathbb{R}^m \)

Again, depends on whether \( b \in \text{col}(A) \), so we can only find “closest” or “least squares” solution. \( x' = A^\dagger b \)

Pseudoinverse

\( A^\dagger \) solves \( Ax = b \) in least squares sense, i.e. \( \|Ax - b\|_2 \) is minimum.
Figure 1: A singular matrix \( A \) has \( \text{Col}(A) \subset \mathbb{R}^n \). This is represented by a plane in the diagram. If \( b \) lies outside of \( \text{Col}(A) \), then the best one can do is to obtain \( b' \), which is the vector in \( \text{Col}(A) \) that is closest to \( b \). This is what the pseudoinverse computes: \( b' = Ax' \), where \( x' = A^\dagger b \).

\[
A^\dagger = V\Sigma^\dagger U^\top \quad \text{(using SVD)}
\]
\[
= (A^\top A)^{-1} A^\top \quad \text{but this requires rank}(A) = n
\]

Note: \( A^\dagger A = (A^\top A)^{-1} A^\top = I \), but \( AA^\dagger = A (A^\top A)^{-1} A^\top \neq I \) in general. Thus, pseudoinverse is only a left inverse, not a right inverse.

If \( A \) invertible, then pseudoinverse = true inverse:

\[
A^\dagger = (A^\top A)^{-1} A^\top
\]
\[
= A^{-1}A^{-\top}A^\top = A^{-1}
\]

In Matlab, always use \( A \backslash b \) to solve \( Ax = b \). "\( \backslash \)" will compute \( A^{-1} \) or \( A^\dagger \) accordingly.

**Matrix Inversion Formulas**


1. Lemma 1 (Inverse of a Partitioned Matrix)

   Let \( R \) denote the partitioned matrix

   \[
   R = \begin{bmatrix} A & B \\ C & D \end{bmatrix}
   \]

   The inverse of \( R \) is

   \[
   R^{-1} = \begin{bmatrix} E^{-1} & FH^{-1} \\ H^{-1}G & H^{-1} \end{bmatrix}
   \]
\[ E = A - BD^{-1}C \]
\[ AF = -B \]
\[ GA = -C \]
\[ H = D - CA^{-1}B \]

All indicated inverses are assumed to exist. The matrix \( E \) is called Schur complement of \( A \), and the matrix \( H \) is called the Schur complement of \( D \).

2. Lemma 2 (Matrix Inversion Lemma)

Let \( E \) denote the Schur complement of \( A \):

\[ E = A - BD^{-1}C \]

Then the inverse of \( E \) is

\[ E^{-1} = A^{-1} + FH^{-1}G \]
\[ AF = -B \]
\[ GA = -C \]
\[ H = D - CA^{-1}B \]

Lemmas 1 and 2 combine to form the following representation for the inverse of a partitioned matrix.

**Theorem (Partitioned Matrix Inverse)**

The inverse of the partitioned matrix

\[
\begin{bmatrix}
A & B \\
C & D
\end{bmatrix}
\]

is the matrix

\[
\begin{bmatrix}
A^{-1} & 0 \\
0 & 0
\end{bmatrix} + \begin{bmatrix}
\frac{F}{T} \\
\frac{H}{I}
\end{bmatrix} [H^{-1}] \begin{bmatrix}
G \\
I
\end{bmatrix}
\]

\[ AF = -B \]
\[ GA = -C \]
\[ H = D - CA^{-1}B \]
Corollary: Woodbury’s Identity

The inverse of the matrix

\[ R = R_0 + \gamma^2 uu^\top \]

is the matrix

\[ R^{-1} = R_0^{-1} - \frac{\gamma^2}{1 + \gamma^2 u^\top R_0^{-1} u} R_0^{-1} uu^\top R_0^{-1} \]

Projections

Often we want to project \( x \) onto some subspace, i.e. find \( y \) in subspace, “closest” to \( x \). Geometrically, this occurs when \( x - y \) is orthogonal to subspace. Often the subspace of interest is \( \text{Col}(A) \). Recall that in the SVD of \( A \), \( U_r \) form an orthogonal basis for \( \text{Col}(A) \).

The projection matrix \( P_A \) that projects any vector onto \( \text{Col}(A) \) is:

\[ P_A = U_r U_r^\top \quad \text{(SVD)} \]

\[ = A \left( A^\top A \right)^{-1} A^\top \]

e.g. To project onto a line (vector) \( u \), \( P_u = \frac{uu^\top}{u^\top u} \).

In general, a projection matrix \( P \) is one that satisfies:

1. \( P^\top = P \) symmetric
2. \( P^2 = P \) idempotent

What are the eigenvalues of \( P \)?

Derivatives

<table>
<thead>
<tr>
<th>w.r.t</th>
<th>Differentiate</th>
<th>scalar</th>
<th>vector</th>
<th>matrix</th>
</tr>
</thead>
<tbody>
<tr>
<td>scalar</td>
<td>scalar</td>
<td>vector</td>
<td>matrix</td>
<td></td>
</tr>
<tr>
<td>vector</td>
<td>vector</td>
<td>matrix</td>
<td></td>
<td></td>
</tr>
<tr>
<td>matrix</td>
<td>matrix</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

scalar—scalar: e.g. \( \frac{d}{dx} x^2 = 2x \)

vector—scalar: e.g.,

\[ y = \begin{bmatrix} \cos \theta & \sin^2 \theta \end{bmatrix}^\top \]

\[ \frac{dy}{d\theta} = \begin{bmatrix} -\sin \theta & 2 \sin \theta \cos \theta \end{bmatrix}^\top \]

matrix—scalar: e.g.,

\[ A = \begin{bmatrix} x^2 & x \\ 1 & \frac{1}{x} \end{bmatrix} \]

\[ \frac{dA}{dx} = \begin{bmatrix} 2x & 1 \\ 0 & -\frac{1}{x^2} \end{bmatrix} \]
scalar—vector: \( f(x) \) scalar function of vector

\[
    x = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}
\]

\[
    \frac{df}{dx} = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{bmatrix}
\]

vector—vector: \( y(x) \) \( m \times 1 \) vector function of vector \( x \in \mathbb{R}^n \)

Then,

\[
    \frac{dy}{dx} = \begin{bmatrix} \frac{\partial y_1}{\partial x_1} & \cdots & \frac{\partial y_m}{\partial x_1} \\ \vdots & \ddots & \vdots \\ \frac{\partial y_1}{\partial x_n} & \cdots & \frac{\partial y_m}{\partial x_n} \end{bmatrix}
\]

\[
y : m \times 1 \]
\[
x : n \times 1
\]

\[
    \frac{dy}{dx} : n \times m \text{ matrix}
\]

scalar—matrix: \( f(A) \) scalar function of \( m \times n \) \( A \)

Then,

\[
    \frac{df}{dA} = \begin{bmatrix} \frac{\partial f}{\partial a_{11}} & \cdots & \frac{\partial f}{\partial a_{1n}} \\ \vdots & \ddots & \vdots \\ \frac{\partial f}{\partial a_{m1}} & \cdots & \frac{\partial f}{\partial a_{mn}} \end{bmatrix} \quad m \times n \text{ matrix}
\]

Commonly used derivatives

1. \( \frac{d}{dx} (Ax) = A^\top \)

2. \( \frac{dx}{dx} = I \)

3. \( \frac{dy^\top x}{dx} = \frac{dx^\top y}{dx} = y \)

4. \( \frac{d}{dx} (x^\top Ax) = \begin{cases} (A + A^\top) x & \text{if } A \text{ square} \\ 2Ax & \text{if } A \text{ symmetric} \end{cases} \)

5. \( \frac{d}{dx} (u^\top(x) \ v(x)) = \left[ \frac{du^\top}{dx} \right] v + \left[ \frac{dv^\top}{dx} \right] u \quad \text{“product rule”} \)

6. \( \frac{d}{dA} \text{tr}(A) = I \)
7. \( \frac{d}{dA} \det(A) = \det(A)A^{-\top} \)

Example: to find pseudoinverse. Let \( e = Ax - b \). We want \( x \) such that \( ||e||_2 \) smallest., i.e. \( ||e||_2^2 \) smallest.

Let \( y = ||e||_2^2 \)
\( = e^\top e \)
\( = (Ax - b)^\top (Ax - b) \)
\( = x^\top A^\top Ax - 2b^\top Ax + b^\top b \)

\( \frac{dy}{dx} = 2A^\top Ax - 2A^\top b = 0 \)
\( \Rightarrow A^\top Ax = A^\top b \)
\( \Rightarrow x = \left( A^\top A \right)^{-1} A^\top b \)

**Hessian: 2nd derivative**

Let \( f(x) \) be scalar function of \( x \in \mathbb{R}^n \)

Then Hessian:

\[
H = \frac{d^2 f}{dx^2} = \begin{bmatrix}
\frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\
\frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\
\vdots & \vdots & \ddots & \vdots \\
\frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2}
\end{bmatrix}
\]

Hessian is symmetric.

**Positive semi-definite (psd)**

A square matrix \( A \) is positive semi-definite if \( x^\top Ax \geq 0 \) for all \( x \neq 0 \). Positive definite \( x^\top Ax > 0 \)

Note: \( A \) is a psd means all eigenvalues \( \geq 0 \).

If a Hessian matrix is psd, then \( f \) has minimum point.

e.g. in the pseudoinverse calculation, \( \frac{dy}{dx} = 2A^\top Ax - 2A^\top b \)

So Hessian, \( H = \frac{d}{dx} \left( \frac{dy}{dx} \right) = 2A^\top A \)

Now, for any \( x \neq 0, x^\top Hx = 2x^\top A^\top Ax = 2||Ax||^2 \geq 0 \) since \( ||Ax||^2 \) is the squared norm.

So \( H \) is psd. \( \Rightarrow y \) has minimum point. This justifies taking derivatives to find best \( x \)