Szymon Rusinkiewicz
COS 302, Fall 2020
Singular Value Decomposition (SVD)

- Matrix decomposition that reveals structure
- Useful for:
  - Inverses, pseudoinverses
  - Stable least-squares, even for unconstrained problems
  - Matrix similarity and approximation
  - Dimensionality reduction and PCA
  - Orthogonalization
  - Constrained least squares and multidimensional scaling

Let’s look at motivation for these
Condition Number

• $\text{cond}(A)$ is function of $A$

• $\text{cond}(A) \geq 1$, bigger is bad

• Measures how change in input propagates to output:

\[
\frac{\| \Delta x \|}{\| x \|} \leq \text{cond}(A) \frac{\| \Delta A \|}{\| A \|}
\]

  – E.g., if $\text{cond}(A) = 451$ then can lose $\log(451) = 2.65$ digits of accuracy in $x$, compared to precision of $A$

• For matrices with real eigenvalues, $\text{cond}(A) = |\lambda_{max}| / |\lambda_{min}|$
Normal Equations are Bad

\[
\frac{\| \Delta x \|}{\| x \|} \leq \text{cond}(A) \frac{\| \Delta A \|}{\| A \|}
\]

• Least squares using normal equations involves solving \( A^T A x = A^T b \)

• \( \text{cond}(A^T A) = [\text{cond}(A)]^2 \)

• E.g., if \( \text{cond}(A) = 451 \) then can lose \( \log(451^2) = 5.3 \) digits of accuracy, compared to precision of \( A \)
Underconstrained Least Squares

• What if you have fewer data points than parameters in your function?
  – Intuitively, can’t do standard least squares
  – Solution takes the form $A^T Ax = A^T b$
  – When $A$ has more columns than rows, $A^T A$ is singular: can’t take its inverse, etc.
Underconstrained Least Squares

- More subtle version: more data points than unknowns, but data poorly constrains function
- Example: fitting to \( y=ax^2+bx+c \)
Underconstrained Least Squares

• Problem: if problem very close to singular, roundoff error can have a huge effect
  – Even on “well-determined” values!

• Can detect this:
  – Uncertainty proportional to covariance $C = (A^T A)^{-1}$
  – In other words, unstable if $A^T A$ has small values
  – More precisely, care if $x^T (A^T A) x$ is small for any $x$

• Idea: if part of solution unstable, set answer to 0
  – Avoid corrupting good parts of answer
Singular Value Decomposition (SVD)

• Handy mathematical technique that has application to many problems
• Given any $m \times n$ matrix $A$, algorithm to find matrices $U$, $V$, and $W$ with:

  $$A = U W V^T$$

  $U$ is $m \times m$ and **orthonormal**
  $W$ is $m \times n$ and **zero except main diagonal**
  $V$ is $n \times n$ and **orthonormal**

• Won’t derive algorithm – treat as black box (e.g., `numpy.linalg.svd`)
“Full” SVD

\[ A = U \begin{bmatrix} w_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & w_n \end{bmatrix} V^T \]

\[ u, w, v^T = \text{numpy.linalg.svd}(a) \]
SVD

• Handwavy explanation: rotate to a basis where all the scaling and stretching of $A$ is along coordinate axes
  – Should remind you of eigendecomposition (which would have $U = V$)
• The $w_i$ are called the **singular values** of $A$
• If $A$ is singular, some of the $w_i$ will be 0
• In general $\text{rank}(A) =$ number of nonzero $w_i$
• SVD is mostly unique (up to permutation of singular values, or if some $w_i$ are equal)
  – The $w_i$ are conventionally returned in sorted order, largest to smallest
Singular Value Decomposition (SVD)

- If $m > n$, only $n$ nonzero rows in $W$, many useless columns in $U$
- If $n > m$, only $m$ nonzero columns in $W$, many useless columns in $V$
  - Define “compact” or “reduced” versions that omit all those zeroes
“Compact” SVD, if $m > n$

$$A = U \begin{bmatrix} w_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & w_n \end{bmatrix} V^T$$

$A \in \mathbb{R}^{m \times n}$, $U \in \mathbb{R}^{m \times n}$, $V \in \mathbb{R}^{n \times n}$,

$$u, w, v^T = \text{numpy.linalg.svd}(a, \text{full_matrices}=False)$$
“Compact” SVD, if $n > m$

$$A = U \begin{bmatrix} w_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & w_m \end{bmatrix} V^T$$

$$u, w, v^T = \text{numpy.linalg.svd}(a, \text{full_matrices}=\text{False})$$
SVD and Inverses

• Why is SVD so useful?

• Application #1: inverses

• \( A^{-1} = (V^T)^{-1} W^{-1} U^{-1} = VW^{-1} U^T \)
  
  – Using fact that inverse = transpose for orthogonal matrices
  
  – Since \( W \) is diagonal, \( W^{-1} \) also diagonal with reciprocals of entries of \( W \)
SVD and the Pseudoinverse

• \( A^{-1} = (V^T)^{-1} W^{-1} U^{-1} = VW^{-1} U^T \)

• This fails when some \( w_i \) are 0
  – It’s *supposed* to fail – singular matrix
  – Happens when rectangular \( A \) is rank deficient

• Pseudoinverse \( A^+ \): if \( w_i = 0 \), set \( 1/w_i \) to 0 (!!)
  – “Closest” matrix to inverse
  – Defined for all (even non-square, singular, etc.) matrices
  – Equal to \( (A^T A)^{-1} A^T \) if \( A^T A \) invertible
SVD and Least Squares

- Solving $Ax = b$ by least squares:
  
  $A^TAx = A^Tb \implies x = (A^TA)^{-1}A^Tb$

- Replace with $A^+$: $x = A^+b$
  
  - Compute pseudoinverse using SVD

- Lets you see if data is singular ($< n$ nonzero singular values)

- Singular values tell you how stable the solution will be
  
  - Condition number = ratio of largest to smallest singular values

- For better stability, set $1/w_i$ to 0 if $w_i$ is small (even if not exactly 0)
  
  - Accuracy / stability tradeoff? Not if that component was underconstrained…
SVD and Matrix Similarity

• One common definition for the norm of a matrix is the Frobenius norm:
  \[ \|A\|_F = \sum_{i} \sum_{j} a_{ij}^2 \]

• Frobenius norm can be computed from SVD
  \[ \|A\|_F = \sum_i w_i^2 \]

• Euclidean (spectral) norm can also be computed:
  \[ \|A\|_2 = \{ \max |\lambda| : \lambda \in \sigma(A) \} \]

• So changes to a matrix can be evaluated by looking at changes to singular values
• Suppose you want to find best rank-$k$ approximation to $A$
• Answer: set all but the largest $k$ singular values to zero
• Can form compact representation by eliminating columns of $U$ and $V$ corresponding to zeroed $w_i$
SVD and Orthogonalization

- $\mathbf{U}$ and $\mathbf{V}$ are orthonormal, all stretching and scaling in $\mathbf{W}$
- The matrix $\mathbf{U}\mathbf{V}^T$ is the “closest” orthonormal matrix to $\mathbf{A}$

  - Yet another useful application of the matrix-approximation properties of SVD
  - Much more stable numerically than Graham-Schmidt orthogonalization
Total Least Squares

- One final least squares application
- Fitting a line: vertical vs. perpendicular error
Total Least Squares

• Distance from point to line:

\[ d_i = \left( \begin{array}{c} x_i \\ y_i \end{array} \right) \cdot \vec{n} - a \]

where \( n \) is normal vector to line, \( a \) is a constant

• Minimize:

\[ \chi^2 = \sum_i d_i^2 = \sum_i \left[ \left( \begin{array}{c} x_i \\ y_i \end{array} \right) \cdot \vec{n} - a \right]^2 \]
Total Least Squares

- First, let’s pretend we know $n$, solve for $a$

\[
\chi^2 = \sum_i \left[ \left( \begin{array}{c} x_i \\ y_i \end{array} \right) \cdot \bar{n} - a \right]^2
\]

\[
a = \frac{1}{m} \sum_i \left( \begin{array}{c} x_i \\ y_i \end{array} \right) \cdot \bar{n}
\]

- Then

\[
d_i = \left( \begin{array}{c} x_i \\ y_i \end{array} \right) \cdot \bar{n} - a = \left( \begin{array}{c} x_i - \frac{\Sigma x_i}{m} \\ y_i - \frac{\Sigma y_i}{m} \end{array} \right) \cdot \bar{n}
\]
Total Least Squares

- So, let’s define

\[
\begin{pmatrix}
\tilde{x}_i \\
\tilde{y}_i
\end{pmatrix} = \begin{pmatrix}
x_i - \frac{\sum x_i}{m} \\
y_i - \frac{\sum y_i}{m}
\end{pmatrix}
\]

and minimize

\[
\sum_i \left[ \left( \tilde{x}_i \cdot \tilde{n} \right)^2 \right]
\]
Total Least Squares

• Write as linear system

\[
\begin{pmatrix}
\tilde{x}_1 & \tilde{y}_1 \\
\tilde{x}_2 & \tilde{y}_2 \\
\tilde{x}_3 & \tilde{y}_3 \\
\vdots & \vdots
\end{pmatrix}
\begin{pmatrix}
n_x \\
n_y
\end{pmatrix} = \tilde{0}
\]

• Have \( An = 0 \)

  – Problem: lots of \( n \) are solutions, including \( n = 0 \)
  – Standard least squares will, in fact, return \( n = 0 \)
Constrained Optimization

- Solution: constrain $n$ to be unit length
- So, try to minimize $\|An\|^2$ subject to $\|n\|^2 = 1$

$$\|An\|^2 = (A\bar{n})^T(A\bar{n}) = \bar{n}^T A^T A \bar{n}$$

- Expand in eigenvectors $e_i$ of $A^T A$:

$$\bar{n} = \mu_1 e_1 + \mu_2 e_2$$

$$\bar{n}^T (A^T A) \bar{n} = \lambda_1 \mu_1^2 + \lambda_2 \mu_2^2$$

$$\|\bar{n}\|^2 = \mu_1^2 + \mu_2^2$$

where the $\lambda_i$ are eigenvalues of $A^T A$
Constrained Optimization

• To minimize $\lambda_1 \mu_1^2 + \lambda_2 \mu_2^2$ subject to $\mu_1^2 + \mu_2^2 = 1$
  set $\mu_{\text{min}} = 1$, all other $\mu_i = 0$

• That is, $n$ is eigenvector of $A^T A$ with the smallest corresponding eigenvalue
SVD and Eigenvectors

• Let $A = UWV^T$, and let $x_i$ be $i^{th}$ column of $V$

• Consider $A^TA x_i$:

$$A^TA x_i = VW^T U^T U W V^T x_i = VW^T V^T x_i = VW^2 1 = V \begin{pmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ w_i^2 \\ \vdots \\ 0 \end{pmatrix} = w_i^2 x_i$$

• So elements of $W$ are $\sqrt{\text{eigenvalues}}$ and columns of $V$ are eigenvectors of $A^T A$
Constrained Optimization

• To minimize $\lambda_1 \mu_1^2 + \lambda_2 \mu_2^2$ subject to $\mu_1^2 + \mu_2^2 = 1$
  set $\mu_{\text{min}} = 1$, all other $\mu_i = 0$

• That is, $n$ is eigenvector of $A^T A$ with
  the smallest corresponding eigenvalue

• That is, $n$ is column of $V$ corresponding to smallest singular value
  – Provides a solution to the total least squares problem
  – Also very related to PCA – next time