Lecture 15, Oct 27, 2023

Interpretations of the SVD

- If $\boldsymbol{A}$ is a linear mapping, we can interpret the SVD as breaking up $\boldsymbol{A}$ into a rotation/projection $\boldsymbol{V}^T$, a scaling $\boldsymbol{\Sigma}$ and reconstructing in the basis $\boldsymbol{U}$
  - The input vector $\boldsymbol{x}$ is projected into the orthonormal basis $\boldsymbol{V}^T$
  - Each component gets scaled by a singular value
  - The rescaled components are reconstructed into an output vector using the basis $\boldsymbol{U}$
  - Note that since the singular values are ordered, $\boldsymbol{v}_1$ is the "highest gain" input vector direction and $\boldsymbol{u}_2$ is the "highest gain" output vector direction
- We can approximate $\boldsymbol{A}$ with a lower rank matrix $\tilde{\boldsymbol{A}}$
  - $\tilde{\boldsymbol{A}}_l = \displaystyle\sum_{i=1}^{l} \sigma_i \boldsymbol{u}_1 \boldsymbol{v}_i^T$
  - Since the singular values are in descending order, we're basically keeping the "more important" parts of the matrix
  - This is linked to dimensionality reduction techniques

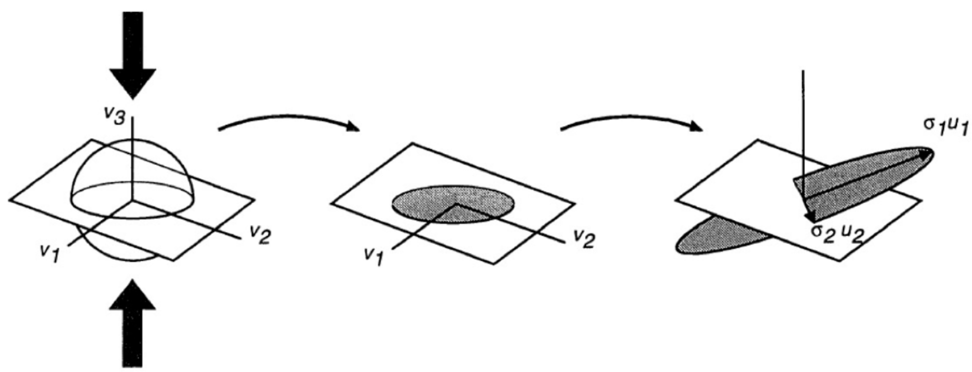

Figure 1: Geometric illustration of the SVD for $n = m = 3$, $k = 2$.

- Geometrically, consider how the SVD transforms a vector on the unit sphere:
  - Let $\boldsymbol{z} = z_1\boldsymbol{v}_1 + z_2\boldsymbol{v}_2 + \cdots + z_n\boldsymbol{v}_n$ for $\displaystyle\sum_i z_i^2 = \boldsymbol{z}^T\boldsymbol{z} = 1$
  - $$\begin{aligned}\boldsymbol{A}\boldsymbol{z} &= \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T\boldsymbol{z} \\ &= \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T(\boldsymbol{z} = z_1\boldsymbol{v}_1 + z_2\boldsymbol{v}_2 + \cdots + z_n\boldsymbol{v}_n) \\ &= \sigma_1 z_1 \boldsymbol{u}_1 + \cdots + \sigma_k z_k \boldsymbol{u}_k \\ &= w_1\boldsymbol{u}_1 + w_2\boldsymbol{u}_2 + \cdots + w_k\boldsymbol{u}_k\end{aligned}$$
  - Since $\displaystyle\sum_{i=1}^{k} \frac{w_i^2}{\sigma_i}^2 = \sum_{i=1}^{k} z_i^2 \leq \sum_{i=1}^{n} z_i^2 = 1$ we have an ellipsoid in k dimensions
    * If $k = n$ (full rank), then we have an equality, so we get the surface of the ellipsoid (no collapse)
    * If $k < n$, we have the inequality so we get the solid interior of the ellipsoid (some dimensions are collapsed)
  - We can interpret the SVD as first collapsing the unit sphere by $n - k$ dimensions, then stretching the remaining k dimensions and then embedding the result in $\mathbb{R}^m$

Applications of the SVD

- SVD has many applications, the most common of which are dimensionality reduction techniques – we can throw away parts of the matrix that are less important

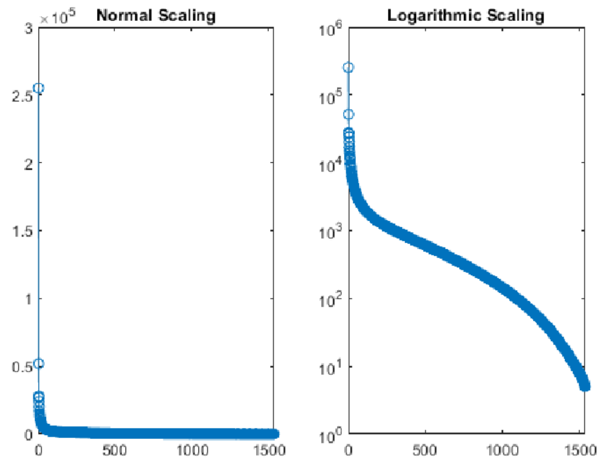

Figure 2: Typical singular value spectrum of an image.

- Image compression: treat the image as a big matrix, take the SVD, and truncate the singular values below a certain threshold
  - In general with data matrices we don't really have a clear "rank", so we will see a "continuous" spectrum of singular values – they decrease relatively smoothly instead of having a cutoff
  - We can now store the singular values and singular vectors above a certain threshold instead of the full image matrix
  - SVD compression is good at picking up patterns that align with the axes of the image (since these correspond to lower rank patterns)
    * e.g. a checkerboard would be very efficient when compressed since it has an effective rank of almost 1; but if we warp this checkerboard so that it is no longer axis-aligned, it becomes worse
- Principal component analysis (PCA): given $\boldsymbol{A}$ as the covariance matrix for a large number of high dimensional data points, we can use an SVD to get a lower dimensional subset that gives us more insight
  - The axes of the SVD are the axes of the error ellipsoid and the singular values are how large the ellipsoid is along each axis
- Dynamic mode decomposition (DMD): finding the best linear operator that represents the nonlinear dynamics of a system: $\dot{\boldsymbol{z}} = \boldsymbol{f}(\boldsymbol{z}) \leftrightarrow \dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x}$
  - The system dynamics are often lifted into higher dimensional space where things become more linear
  - We get *emergent dynamics* from the system, e.g. the behaviour of vortices shed by an object
  - DMD uses a large number of samples of the time series evolution of the dynamics in higher dimensional space: $\{\, \boldsymbol{x}(t_0), \boldsymbol{x}(t_1), \dots, \boldsymbol{x}(t_N) \,\} = \{\, \boldsymbol{x}_0, \boldsymbol{x}_1, \dots, \boldsymbol{x}_N \,\}$
    * Each $\boldsymbol{x}$ is a stacked vector of all the data (state) of the system at a particular time sample
  - Assume there is some matrix $\boldsymbol{A}$ such that $\boldsymbol{x}_{k+1} = \boldsymbol{A}\boldsymbol{x}_k$
    * $\boldsymbol{X'} = \boldsymbol{A}\boldsymbol{X}$ where $\boldsymbol{X'} = \begin{bmatrix}\boldsymbol{x}_1 & \cdots & \boldsymbol{x}_N\end{bmatrix}, \boldsymbol{X} = \begin{bmatrix}\boldsymbol{x}_0 & \cdots & \boldsymbol{x}_{N-1}\end{bmatrix}$
  - Now we need to find $\boldsymbol{A}$, but it can be very large, so we can approximate it with a smaller matrix $\boldsymbol{A}_r$ using the SVD of the data matrix
    * $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T \implies \boldsymbol{X'} = \boldsymbol{A}\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$
    * Truncate the SVD to get the dominant modes: $\boldsymbol{X'} \approx \boldsymbol{A}\boldsymbol{U}_r\boldsymbol{\Sigma}_r\boldsymbol{V}^T$
    * Therefore $\boldsymbol{U}_r^T\boldsymbol{X'}\boldsymbol{V}_r\boldsymbol{\Sigma}_r^{-1} = \boldsymbol{U}_r^T\boldsymbol{A}\boldsymbol{U}_r = \boldsymbol{A}_r$
  - Now we have the transition matrix we can perform an eigendecomposition $\boldsymbol{A}_r\boldsymbol{W} = \boldsymbol{W}\boldsymbol{\Lambda}$ to look at its modes
    * Map the eigenspace back to the original space: $\boldsymbol{\Phi} = \boldsymbol{X'}\boldsymbol{V}_r\boldsymbol{\Sigma}_r^{-1}\boldsymbol{W}$
    * The eigenvalues corresponding these modes allow us to see how they evolve – whether they

grow or shrink with time, oscillations, etc, just like a linear system
  - This can also be used to predict the future steps of the dynamics
- Frame-to-frame visual odometry: finding the coordinate transformation that maps one point cloud to another
  - This can be performed using an SVD of the point cloud data