CS 315 - Day 30, Outer Products, the SVD, and Compression

From last class, when we were presenting outer products:

$$B = \begin{pmatrix} 1 & 4 \\ 3 & 2 \end{pmatrix}, C = \begin{pmatrix} 6 & 5 \\ 8 & 9 \end{pmatrix}$$

If we break B into two column vectors, $\begin{pmatrix} 1 \\ 3 \end{pmatrix}$ and $\begin{pmatrix} 4 \\ 2 \end{pmatrix}$, and C into two row vectors, $(6 \quad 5)$ and $(8 \quad 9)$, which looks like:

$$B = \begin{pmatrix} \uparrow & \uparrow \\ u_1 & u_2 \\ \downarrow & \downarrow \end{pmatrix}, \qquad C = \begin{pmatrix} \leftarrow & v_1 & \rightarrow \\ \leftarrow & v_2 & \rightarrow \end{pmatrix}$$

we can calculate BC using outer products:

$$u_1 v_1 + u_2 v_2$$

$$= \begin{pmatrix} 1 \\ 3 \end{pmatrix} (6 \quad 5) + \begin{pmatrix} 4 \\ 2 \end{pmatrix} (8 \quad 9)$$

$$= \begin{pmatrix} 1 \cdot 6 & 1 \cdot 5 \\ 3 \cdot 6 & 3 \cdot 5 \end{pmatrix} + \begin{pmatrix} 4 \cdot 8 & 4 \cdot 9 \\ 2 \cdot 8 & 2 \cdot 9 \end{pmatrix}$$

$$= \begin{pmatrix} 6 & 5 \\ 18 & 15 \end{pmatrix} + \begin{pmatrix} 32 & 36 \\ 16 & 18 \end{pmatrix}$$

$$= \begin{pmatrix} 38 & 41 \\ 34 & 33 \end{pmatrix}$$

Note, the exact same calculations happen as above, just in different order, which still leads to the same result.

Consider the two intermediate matrices calculated: $\begin{pmatrix} 6 & 5 \\ 18 & 15 \end{pmatrix}$ and $\begin{pmatrix} 32 & 36 \\ 16 & 18 \end{pmatrix}$.

Can you find some multiple for each, relating one row (or column) to the other row (or column)?

If rows (or columns) are related like that, it affects the rank of the matrix. And these intermediate matrices, from the outer product of two vectors, will always have rank 1, where the rank represents the number of unique dimensions. Since one row (or column) is a multiple of the other, the rank is 1 instead of 2.

Definition The *singular value decomposition* of a real $m \times n$ matrix A of rank r has the form $A = UDV^T$ where U and V are orthogonal matrices of sizes $m \times m$ and $n \times n$, respectively, and D is a diagonal matrix having the same dimensions as A. If U and V have columns $[\mathbf{u}_1 \quad \mathbf{u}_2 \quad \cdots \quad \mathbf{u}_m]$ and $[\mathbf{v}_1 \quad \mathbf{v}_2 \quad \cdots \quad \mathbf{v}_n]$ then $\mathbf{u}_i$ is the i^{th} *left singular vector*, and $\mathbf{v}_i$ is the right singular vector. The nonzero diagonal entries of D are called *singular values*; they can be rearranged so $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0$. The equation $A = UDV^T$ is equivalently expressed as

$$A = \sigma_1 \mathbf{u}_1 \mathbf{v}_1^T + \sigma_2 \mathbf{u}_2 \mathbf{v}_2^T + \cdots + \sigma_r \mathbf{u}_r \mathbf{v}_r^T. \qquad (*)$$

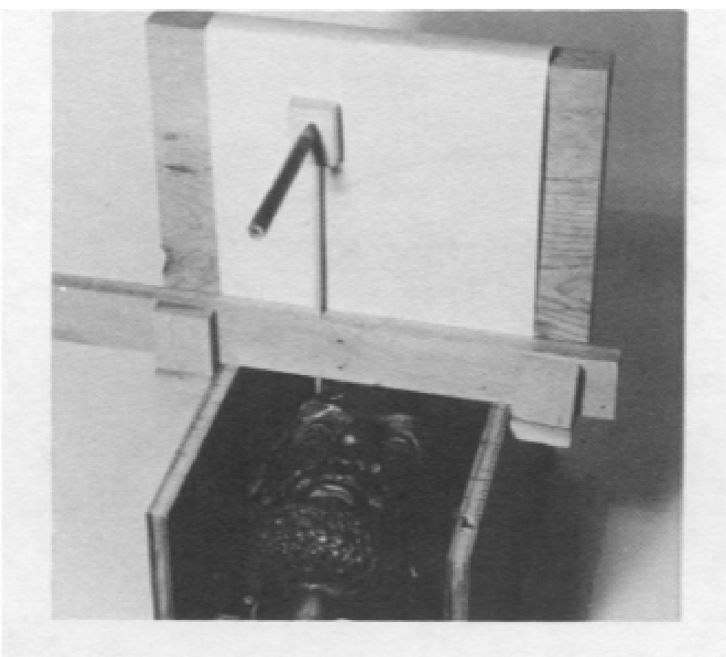
So we can decompose A into three matrices $U, D, and\ V$, where U and V are comprised of vectors. It's beyond the scope of this course, but it's interesting to note that U and V^T are orthonormal, their transposes are equal to their inverses!

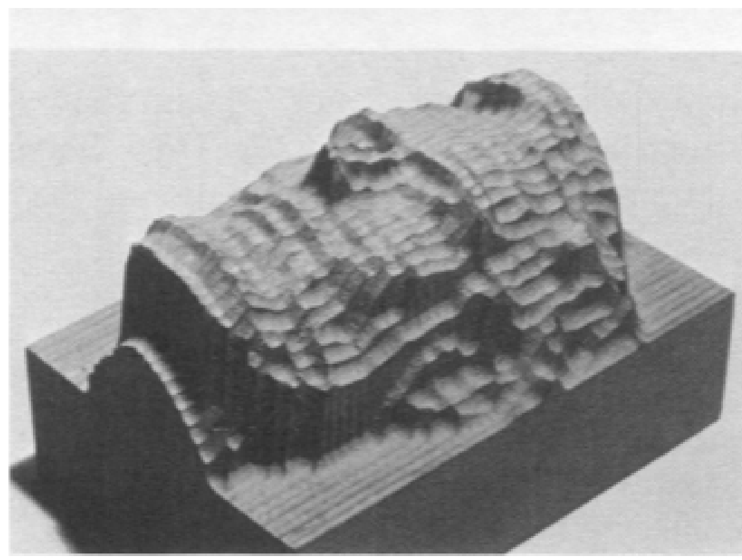
The most important property that $\mathbf{u}_1$, $\mathbf{v}_1$, and σ_1 possess is that they combine to make $\sigma_1 \mathbf{u}_1 \mathbf{v}_1^T$ the closest possible rank-one approximation to A. "Closest possible" means that among all possible unit vectors $\mathbf{u}$ and $\mathbf{v}$ and all possible scalars c, the norm of the difference $A - c\mathbf{u}\mathbf{v}^T$ is the smallest possible, where the norm, like for vectors, is the square root of the sum of the squares of the entries of the matrix.

$\mathbf{u}_1$ is a vertical $m \times 1$ vector and $\mathbf{v}_1^T$ is a horizontal $1 \times n$ vector, so the outer product $\mathbf{u}_1 \mathbf{v}_1^T$ has dimensions $m \times n$, the same as A. Every column of $\sigma_1 \mathbf{u}_1 \mathbf{v}_1^T$ is a multiple of $\mathbf{u}_1$, so in this sense $\mathbf{u}_1$ gives the most typical or important profile vertically. Analogously $\mathbf{v}_1^T$ gives a most important profile horizontally. Since $\mathbf{u}_1$ and $\mathbf{v}_1^T$ are unit vectors, a final step is needed of scaling $\mathbf{u}_1 \mathbf{v}_1^T$ up or down by the number σ_1 to achieve the closest possible match to A.

Ok, there is a LOT to parse here, but a physical example first, might be helpful.

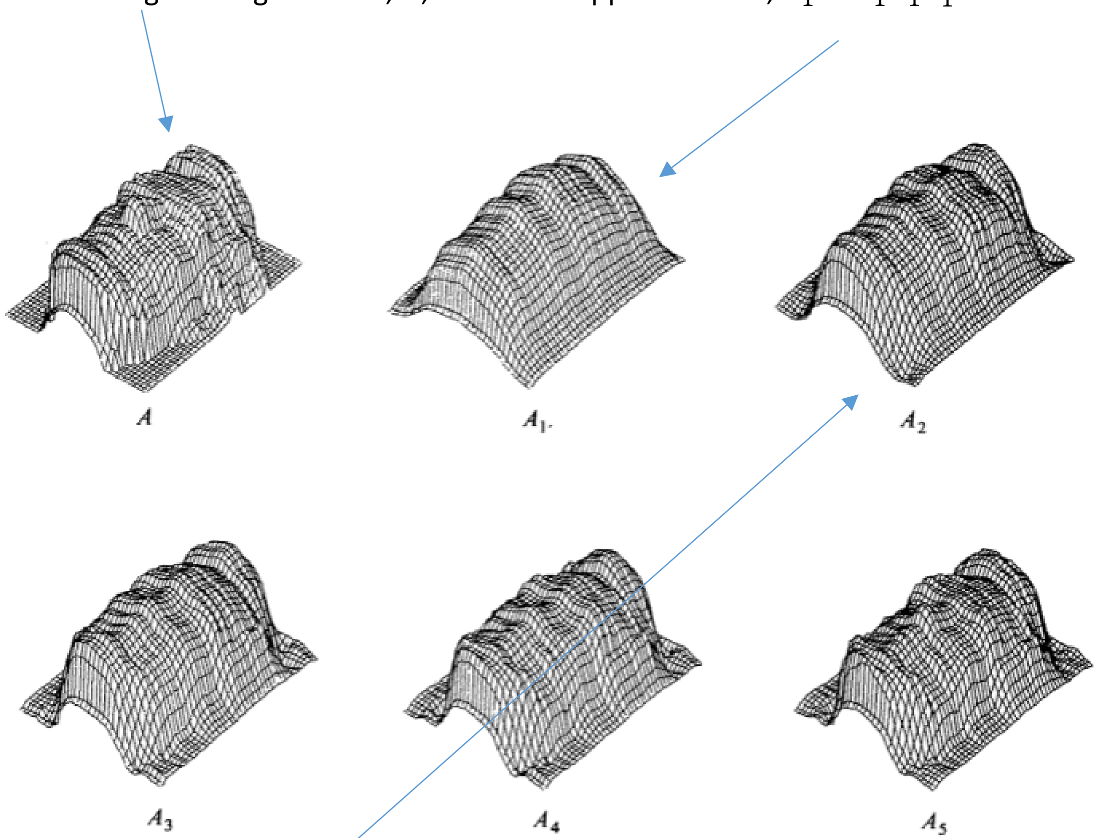
In the 1980s, before scanning, mathematicians took measurements of the height of the surface of a bust of Abraham Lincoln, and then the pixelated recreation:





These height values are just numeric entries in a 2D (49×36) array → a matrix we can calculate the SVD for!

Here are the original height values, A, the rank 1 approximation, $A_1 = \sigma_1 \boldsymbol{u}_1 \boldsymbol{v}_1^T$



The rank 2 approximation, $A_2 = \sigma_1 \boldsymbol{u}_1 \boldsymbol{v}_1^T + \sigma_2 \boldsymbol{u}_2 \boldsymbol{v}_2^T$, and so on…

The most prominent features are displayed in A_1, which corresponds to the largest Singular Value σ_1.

The rank 5 approximation, $A_5 = \sigma_1 \boldsymbol{u}_1 \boldsymbol{v}_1^T + \sigma_2 \boldsymbol{u}_2 \boldsymbol{v}_2^T + \cdots + \sigma_5 \boldsymbol{u}_5 \boldsymbol{v}_5^T$ is actually a not bad approximation, and created with much less information!

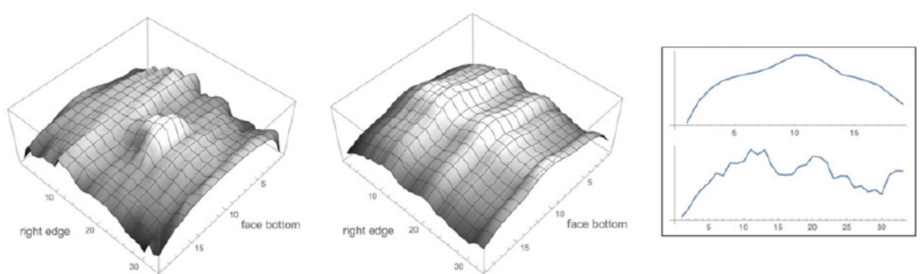


Figure 3. Human Face (Abe) $\sigma_1 \mathbf{u}_1 \mathbf{v}_1^T$ (rk-one approx.) $\mathbf{v}_1$ (top) and $\mathbf{u}_1$ (bottom).

One can read and interpret the profiles of $\mathbf{v}_1$ and $\mathbf{u}_1$ in the rightmost panel of Figure 3 in terms of the left and central panels. The human face (the original) has, for most crosscuts that run from the right side of the face to the left, a common low/high/low profile. This commonality is reflected in the entry-by-entry profile plot of $\mathbf{v}_1$. However for $\mathbf{u}_1$, the analogous top-to-bottom face commonality, although present, is considerably less clear. Nevertheless there is a good alternate way to visualize $\mathbf{u}_1$. Note in Figure 3 the leftmost entries of $\mathbf{u}_1$ are near zero, meaning the top row of $\sigma_1 \mathbf{u}_1 \mathbf{v}_1^T$ (Abe's forehead), is a very small multiple of $\mathbf{v}_1^T$. As one moves across the profile plot of $\mathbf{u}_1$ (that is, vertically down Abe's forehead) the values of $\mathbf{u}_1$ rise, indicating a higher multiple of the $\mathbf{v}_1^T$ profile. Then $\mathbf{u}_1$ drops a bit at eye level, then rises again at the nose level, then drops for the mouth, and finally rises just at the end for the beard. Combining the effects of $\mathbf{u}_1$ with those of $\mathbf{v}_1$ and scaling by the number σ_1 results in the surface $\sigma_1 \mathbf{u}_1 \mathbf{v}_1^T$, the center panel of Figure 3, which displays the best rank-one approximation to the original matrix.

σ_1 is the largest singular value, and from above it is related to the largest features of the original matrix.

Next we will explore a visualization of it.

Conceptualizations for small matrices: the greatest stretch visualizations for $\mathbf{u}_1$, $\mathbf{v}_1$, $\mathbf{u}_2$, $\mathbf{v}_2$

Real 2×2 matrices have an easily understood visualization for their singular vectors. When $A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$ is considered as a mapping of $\mathbb{R}^2$ to $\mathbb{R}^2$ via $\begin{bmatrix} a & b \\ c & d \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \alpha \\ \beta \end{bmatrix}$, the unit circle is mapped onto an ellipse (Figure 1). The direction of greatest stretch of the ellipse gives the direction of the unit vector $\mathbf{u}_1$ (see [4, pp. 5–6] and [10, p. 392]). Furthermore σ_1 is the length of the longest semi-axis, and $\mathbf{v}_1$ is the unit vector such that $A\mathbf{v}_1 = \sigma_1\mathbf{u}_1$. These constitute useful and pleasant visualizations of $\mathbf{u}_1$, $\mathbf{v}_1$, and σ_1. Note that the greatest stretch occurs in two opposite directions, reflecting the $\pm$ ambiguity of singular vectors. Second singular vectors are also visually obvious. Since $\mathbf{v}_2$ is a unit vector orthogonal to $\mathbf{v}_1$, its dotted-in direction is clear. Similarly $\mathbf{u}_2$ is the dotted unit vector orthogonal to $\mathbf{u}_1$ in the image space, and σ_2 is the length of its semi-axis.

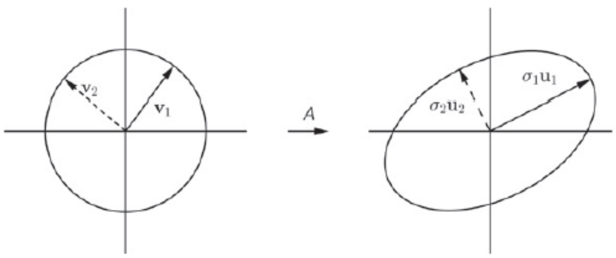


Figure 1. For 2×2 matrices; $\mathbf{u}_1$ lies in the direction of greatest stretch, and $\mathbf{v}_1$ is the pre-image of $\sigma_1\mathbf{u}_1$.

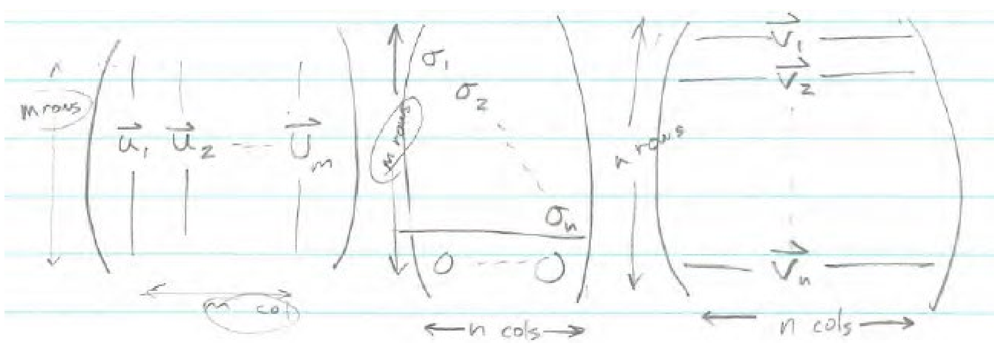
First, read up on the theory behind interactive SVD visualizer, https://github.com/krusegw/krusegw.github.io/blob/main/cs315/Intro_to_the_use_of_the_GUI.pdf

Then, try the interactive SVD visualizer, https://krusegw.github.io/cs315/james_downloadable_GUI.html

So, another application of SVD is image compression. Just like for the bust of Abraham Lincoln, we can take a 2D array of pixel values (a matrix!) and calculate the SVD. Then we can truncate the number of slices, capturing the main features of the image using much less information.

The SVD is $A = UDV^T$, where A is of rank r, U is size $m \times m$ and V is $n \times n$, and D is a diagonal matrix the same size as A.

$$A_{m\times n} = U_{m\times m}\, D_{m\times n}\, V^T_{n\times n}$$



Note in the diagram, D might instead have more columns than rows, leading to 0 columns.

The small singular values, getting closer to σ_m (or σ_n) come from the “boring” or “small” features, and can be ignored.

$$A = \sigma_1 \boldsymbol{u}_1 \boldsymbol{v}_1^T + \sigma_2 \boldsymbol{u}_2 \boldsymbol{v}_2^T + \cdots + \sigma_r \boldsymbol{u}_r \boldsymbol{v}_r^T$$

becomes:

$$A = \sigma_1 \boldsymbol{u}_1 \boldsymbol{v}_1^T + \sigma_2 \boldsymbol{u}_2 \boldsymbol{v}_2^T + \cdots + \sigma_k \boldsymbol{u}_k \boldsymbol{v}_k^T$$

which in matrix notation is:

$$A_{m\times n} = U_{m\times k}\, D_{k\times k}\, V_{k\times n}^T$$

The approximation to original can be constructed by multiplying everything out, but for the actual compression, the first k singular values and the first k vectors u and v would be saved.

So, if $A_{340\times 280}$, it has $340 * 280 = 95{,}200$ elements.

To compress using $k = 20$, the u vectors would have 340 elements, and there would be 20 of them, $340 * 20 = 6{,}800$, the u vectors would have 280 elements, and there would be 20 of them, $280 * 20 = 5{,}600$, so $6{,}800 + 5{,}600 + 20 = 12{,}420$ numeric values would need to be stored, just 13% of the numeric values in the original image.

Compression via SVD’s

Original Color JPEG image, 80 x 113

Greyscale image, 80 x 113, created from SVD with ***k=10***

Greyscale image, 80 x 113, created from SVD with ***k=15***

Greyscale image, 80 x 113, created from SVD with ***k=20***

Greyscale image, 80 x 113, created from SVD with ***k=30***

A template for doing compressions is here:
https://github.com/krusegw/svd/blob/main/SVDImageCompression.ipynb

For more info, this module is based on:

- https://tlakoba.w3.uvm.edu/AppliedUGMath/for_talks/InfoRetrieval/1983_Visualization_SVD.pdf
- https://www.tandfonline.com/doi/full/10.1080/07468342.2023.2201567
- https://understandinglinearalgebra.org/chap7.html