Project 2: Singular Value Decomposition

Ryan Hausner

October 26th, 2024

1 Part 1: Intro to SVD

(a) Describe the three matrices that make up SVD as described in the video.

The three matrices that make up SVD as described in the video are U, Σ, and V^T. Each of these matrices has multiple interpretations. The first interpretation and arguably the most important is the geometric interpretation. The matrices together act as a way to rewrite an $m \times n$ matrix A which can be rectangular, as the product of three matrices. It is a way to generalize the transformation of a matrix to simpler matrices.

(a) V^T: This matrix is the transpose of the matrix U which is the matrix of normalized eigenvectors of AA^T.
  - The geometric interpretation of this is a rotation back to the standard basis of the right singular vectors. In $\mathbb{R}^2$ it applies a rotation back to the x and y axis. The largest singular value σ lies on the x axis.

(b) Σ: The second matrix Σ is a diagonal matrix with the singular values $\sigma_1, \sigma_2, ...\sigma_n$ along the main diagonal such that $\sigma_1 \geq \sigma_2 \geq ...\sigma_n \geq 0$. These singular values are the square roots of the eigenvalues, $\lambda_1, \lambda_2, ..., \lambda_n$.
  - The geometric meaning of this matrix is a transformation that stretches the x and y axis and removes a dimension. For example, it can send a matrix from $\mathbb{R}^3 \to \mathbb{R}^2$.

(c) U: This matrix is a matrix of normalized eigenvectors of the original matrix A (Left Singular Vectors).
  - The geometric meaning of this matrix is a rotation to align with the axis of the space.

The composition of these three matrices, $S\Sigma V^T$ applies the same transformation that the matrix A does.

(b) Describe how what the slider on the image compression website controls.

The slider on the image compression website controls the rank of the approximation in the SVD algorithm. It controls how many singular values are used for the approximation, using the largest ones first. In the matrix Σ, the singular values σ are in descending order along the diagonal. Because of this, the smaller k value yields the highest singular value which in turn returns the most information. As k increases, the more singular values that are included and the closer it gets to generating the original image. By using a low-rank approximation, an image is able to be expressed using significantly less data. By increasing the rank of the approximation, the original image is able to be shown using less data.

(c) Using the image compression demo, choose three images (or you can upload your own on the website), write down the dimensions of the image and the approximate value of k from the slider that you believe maintains a near-perfect compressed version of the image.

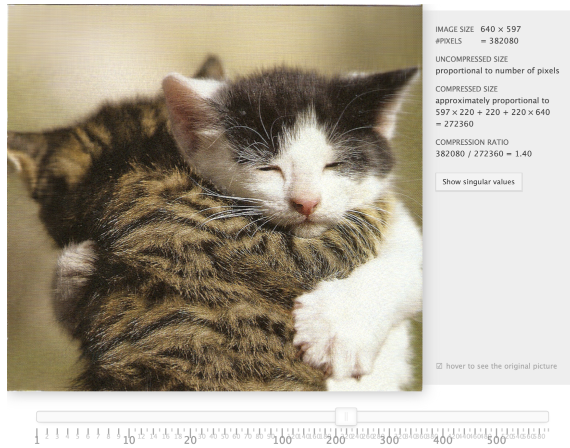

Figure 1: Image compression of cats.

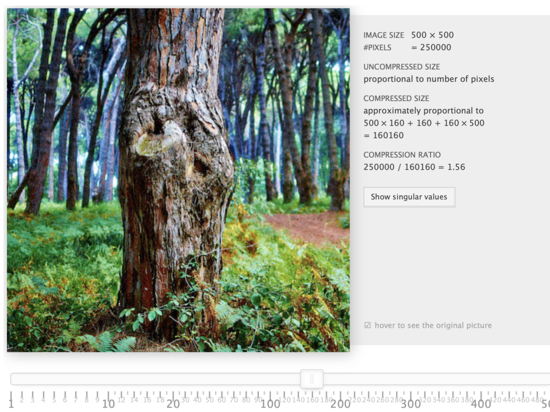

Figure 2: Image compression of a tree.

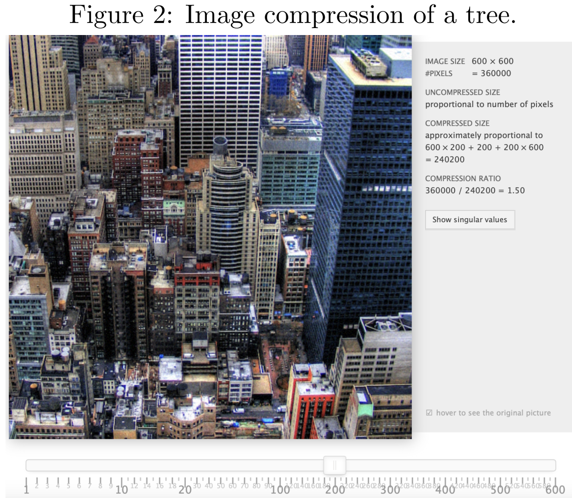

Figure 3: Image compression of a city.

Table 1: Image Compression data

Name	Size	Pixels	k-value
Cats	640×597	382080	220
Tree	500×500	250000	160
City	600×600	360000	200

(d) Do you notice any patterns from your analysis in the previous part?

A pattern from the previous part is the smaller the number of pixels in an image, the smaller the k-value needed to generate a near-perfect image. Another pattern found from the k-value scale on the compression website is that when k is closer to 0, there is a bigger difference in change than further up on the scale. As the image becomes closer to its original version, the singular values asymptote and become less impactful.

(e) Prove that for any $m \times n$ matrix A, AA^T and A^TA are symmetric matrices.

For a matrix A to be symmetric, $A = A^T$.

$$\begin{gathered}\text{Starting with } AA^T\text{:}\\ (AA^T)^T = AA^T\\ (A^T)^T(A^T) = AA^T\\ AA^T = AA^T\end{gathered}$$

Therefore AA^T is symmetric. Similarly for A^TA:

$$\begin{gathered}(A^TA)^T = A^TA\\ (A^T)(A^T)^T = A^TA\\ A^TA = A^TA\end{gathered}$$

(f) Using the following steps, prove that if A is a 2×2 symmetric matrix then its eigenvectors are perpendicular by following these steps. Hint: Recall the dot product can be viewed as matrix multiplication $\vec{v} \cdot \vec{w} = \vec{v}^T\vec{w}$

(a) Consider an arbitrary 2×2 symmetric matrix $A = \begin{bmatrix} a & b \\ b & c \end{bmatrix}$.[1] Show that A is non-defective.

A matrix A is non-defective if it has n distinct eigenvalues.
Finding the eigenvectors of A:
$\det(A - \lambda I_n) = \begin{vmatrix} a-\lambda & b \\ b & c-\lambda \end{vmatrix} = (a-\lambda)(c-\lambda) - b^2 = 0$
Using Wolfram Alpha the roots of the characteristic polynomial $p(\lambda)$ are

$$\lambda = \pm\sqrt{a^2 - 2ac + 4b^2 + c^2} + a + c$$

To ensure two distinct eigenvalues, the discriminant must be > 0: $a^2 - 2ac + 4b^2 + c^2 > 0$.

$$(a-c)^2 + 4b^2 > 0$$

As long as $a \neq c$ & $b \neq 0$, the matrix A is non-defective.

[1]Note the matrix is already diagonal if it is in the form $\begin{bmatrix} a & 0 \\ 0 & a \end{bmatrix}$ and is therefore non-defective.

(b) Let $\vec{v_1}$ and $\vec{v_2}$ be two eigenvectors of A with eigenvalues λ_1 and λ_2 (where $\lambda_1 \neq \lambda_2$). Simplify the following expression in two ways

$$(A\vec{v_1}) \cdot \vec{v_2}$$

(1) The first way to simplify this expression is to recall a key aspect of eigenvalues: $A\vec{v} = \lambda\vec{v}$.

Substituting $A\vec{v_1}$ for $\lambda_1\vec{v_1}$ produces:

$$\lambda_1(\vec{v_1} \cdot \vec{v_2})$$
$$(A\vec{v_1}) \cdot \vec{v_2} = \lambda_1(\vec{v_1} \cdot \vec{v_2})$$

(2) The second way involves using $\vec{v} \cdot \vec{w} = \vec{v}^T\vec{w}$:

$$A\vec{v_1} \cdot \vec{v_2} = (\vec{v_1}^T A^T)(\vec{v_2})$$

Because A is a symmetric matrix, $A = A^T$.

Simplifying: $\vec{v_1}^T(A\vec{v_2})$

Substituting $A\vec{v_2} = \lambda_2\vec{v_2}$: $\vec{v_1}^T(\lambda_2\vec{v_2})$

Using the same property of dot product from before ($\vec{v} \cdot \vec{w} = \vec{v}^T\vec{w}$):

$$\lambda_2(\vec{v_1} \cdot \vec{v_2})$$
$$(A\vec{v_1}) \cdot \vec{v_2} = \lambda_2(\vec{v_1} \cdot \vec{v_2})$$

(c) Use your answers in (b) to conclude that $\vec{v_1} \cdot \vec{v_2} = 0$

Because we know that $\lambda_1 \neq \lambda_2$, the only way for $\lambda_1(\vec{v_1} \cdot \vec{v_2}) = \lambda_2(\vec{v_1} \cdot \vec{v_2})$ is if $\vec{v_1} \cdot \vec{v_2} = 0$

This tells us that the eigenvectors are perpendicular.

2 Part Two: Singular Value Decomposition

Poll three friends and write down a 4×5 matrix (like the 4×5 movie preference matrix in the video) with their and your preferences for each title on a scale from 0 - 5 where 0 means they have not seen or heard of the title before, 1, means they hate it and 5 means they love it. Be sure to label the rows and columns of your matrix.

The category I chose was musical artists. The artists I chose are Taylor Swift, SZA, Katy Perry, the 1975, and the Beatles.

I polled three friends and created the data matrix A.

Name	Taylor Swift	SZA	Katy Perry	1975	Beatles
Palak	1	3	2	0	4
Maya	1	4	2	0	5
Nick	3	5	0	0	4
Ryan	3	4	2	2	4

$$A = \begin{bmatrix} 1 & 3 & 2 & 0 & 4 \\ 1 & 4 & 2 & 0 & 5 \\ 3 & 5 & 0 & 0 & 4 \\ 3 & 4 & 2 & 2 & 4 \end{bmatrix}$$

(a) Using technology, compute the singular value decomposition of your matrix and list the three matrices here. You may round any decimals to two places.

Using MATLAB, the SVD algorithm was run. The resulting matrices were produced.

$$U = \begin{bmatrix} 0.42 & -0.48 & -0.04 & -0.77 \\ 0.52 & -0.53 & -0.24 & 0.63 \\ 0.53 & 0.65 & -0.54 & -0.09 \\ 0.53 & 0.25 & 0.81 & 0.08 \end{bmatrix}$$

$$\Sigma = \begin{bmatrix} 12.79 & 0 & 0 & 0 & 0 \\ 0 & 2.62 & 0 & 0 & 0 \\ 0 & 0 & 2.09 & 0 & 0 \\ 0 & 0 & 0 & 0.28 & 0 \end{bmatrix}$$

$$V^T = \begin{bmatrix} 0.32 & 0.64 & 0.25 & -0.62 & 0.21 \\ 0.63 & 0.26 & -0.26 & 0.28 & -0.62 \\ 0.23 & -0.58 & 0.50 & -0.43 & -0.41 \\ 0.08 & 0.19 & 0.77 & 0.59 & 0.10 \\ 0.66 & -0.38 & -0.14 & 0.10 & 0.62 \end{bmatrix}$$

(b) You will do an estimate with $k = 2$, meaning you take only the first two columns of U, upper 2×2 matrix of Σ (first two singular values), and first two rows of V^T . Compute their product and write it here.

Creating an estimate of the original matrix A using a low-rank ($k = 2$) involves creating modified matrices U, Σ, and V^T which can be denoted as U', Σ', and $V^{T'}$.

$$U' = \begin{bmatrix} 0.42 & -0.48 \\ 0.52 & -0.53 \\ 0.53 & 0.65 \\ 0.53 & 0.25 \end{bmatrix}$$

$$\Sigma' = \begin{bmatrix} 12.79 & 0 \\ 0 & 2.62 \end{bmatrix}$$

$$V^{T'} = \begin{bmatrix} 0.32 & 0.64 & 0.25 & -0.62 & 0.21 \\ 0.63 & 0.26 & -0.26 & 0.28 & -0.62 \end{bmatrix}$$

Computing the product:

$$U\Sigma V^T = \begin{bmatrix} 0.93 & 3.11 & 1.67 & -3.68 & 1.91 \\ 1.25 & 3.90 & 2.02 & -4.51 & 2.26 \\ 3.24 & 4.78 & 1.25 & -3.73 & 0.38 \\ 2.58 & 4.51 & 1.52 & -4.02 & 1.02 \end{bmatrix}$$

By using a low-rank approximation of $k = 2$, SVD is able to take the two largest singular values ($\sqrt{\lambda}$), which give insight into how the matrix is transformed. This is able to take a large amount of data and compress it. Using the larger singular values allows representation of matrices (including data matrices) using less computation.

(c) What is the matrix you computed in (b), how does it relate to the original?

The matrix computed in (b) is: $U\Sigma V^T = \begin{bmatrix} 0.93 & 3.11 & 1.67 & -3.68 & 1.91 \\ 1.25 & 3.90 & 2.02 & -4.51 & 2.26 \\ 3.24 & 4.78 & 1.25 & -3.73 & 0.38 \\ 2.58 & 4.51 & 1.52 & -4.02 & 1.02 \end{bmatrix}$

This matrix has the same dimensions as the original matrix A (4×5). The matrix Σ acts as a medium for the SVD and allows for the algorithm to be run with a lower rank approximation, attempting to keep the most significant data (by using the largest singular values, σ). The center matrix Σ matches the columns of U and the rows of V^T, allowing to use a smaller matrix to create an estimate for the original.

For matrix A with $k = 2$, $U = 4 \times 2$, $\Sigma = 2 \times 2$, and $V^T = 2 \times 5$. The resulting matrix retains the dimensions of the original matrix. $4 \times 2 \times 2 \times 2 \times 2 \times 5 = 4 \times 5$.

(d) Choose any of the entries in your matrix that are 0 and ask your friend to check out the appropriate title and provide a rating. How does their rating compare to the associated value in the estimate you found in (b)? Why?

After asking Nick to listen to Katy Perry, he gave her a rating of 3 (out of 5).

The updated matrix: $A' = \begin{bmatrix} 1 & 3 & 2 & 0 & 4 \\ 1 & 4 & 2 & 0 & 5 \\ 3 & 5 & 3 & 0 & 4 \\ 3 & 4 & 2 & 2 & 4 \end{bmatrix}$

Comparing this to the rank-2 approximation: $U\Sigma V^T = \begin{bmatrix} 0.93 & 3.11 & 1.67 & -3.68 & 1.91 \\ 1.25 & 3.90 & 2.02 & -4.51 & 2.26 \\ 3.24 & 4.78 & 1.25 & -3.73 & 0.38 \\ 2.58 & 4.51 & 1.52 & -4.02 & 1.02 \end{bmatrix}$

The rank-2 approximation has entry $A_{32} = 1.25$ compared to $A'_{32} = 3$. While this isn't a super accurate approximation, it is still relatively close considering the amount of data and the rank of the approximation. By using a fundamental idea of linear algebra: the dot product, SVD is able to measure the similarity between vectors. SVD is then able to use the similarity between vectors to make predictions for what an entry would be if it equals 0. Because the other entries in column 2 of the matrix A:

$$\begin{bmatrix} 2 \\ 2 \\ 0 \\ 2 \end{bmatrix}$$

and the entries in the third row of A: $\begin{bmatrix} 3 & 5 & 0 & 0 & 4 \end{bmatrix}$ are relatively similar (especially in the column), SVD can measure the similarity using dot product and use the two largest singular values to stretch/compress the most important data accordingly.

(e) Ask ChatGPT how singular value decomposition relates to generative A.I. and ChatGPT. Write down its response (or your favorite part of its response). Does what it said make sense?

I asked ChatGPT: "How does singular value decomposition relate to generative A.I. and ChatGPT?" and it responded with:

- Singular Value Decomposition (SVD) is a powerful linear algebra technique used in matrix factorization, and it plays a crucial role in various areas of generative AI, including in the underlying processes of models like ChatGPT. Here's a breakdown of how SVD connects to generative AI and large language models:
- Dimensionality Reduction: In natural language processing (NLP) and generative AI, SVD is often used for dimensionality reduction in tasks like latent semantic analysis (LSA). By decomposing matrices, such as word-document matrices, into singular values, SVD captures important patterns and relationships, reducing the dimensionality of data while retaining essential information. This reduced representation allows for more efficient training of generative models, as the data is more compact and manageable.

- Data Compression in Training: SVD helps compress large weight matrices by approximating them with low-rank matrices, which can be useful for reducing computational complexity in training models. Large-scale models like ChatGPT process immense amounts of data and rely on efficient matrix computations, so SVD-based techniques are sometimes used to reduce the number of parameters or to compress pre-trained models without a significant loss of performance.
- While SVD itself isn't directly employed in the latest generative transformer architectures, its principles of matrix decomposition, dimensionality reduction, and data simplification inform a lot of foundational ideas in machine learning, which remain relevant in generative AI's evolution.

The main part of ChatGPT's response to the use of SVD is about dimensionality reduction and data compression. Both of these make sense due to the large amount of data ChatGPT processes, it needs a way to take the most important information. Additionally, due to the large amount of data, it needs a fast way to process it and find patterns and similarities between it. It also points out that SVD isn't "employed in the latest generate transformer architectures", it is still very important in compressing data (similarly to in image compression) and dimensionality reduction.