SINGULAR VALUE DECOMPOSITION AND IMAGE COMPRESSION

Robert Santos Picardo

November 6, 2022

1 Singular value decomposition

Singular value decomposition is a particular way of approximately generalizing the diagonalization of a square diagonalizable matrix into any $m \times n$ matrix (that may be not necessarily diagonalizable). First, we first state the singular value decomposition for any linear transformation between finite-dimensional vector spaces, before we can start working with the matrix variant of the following theorem.

Theorem 1.1 (Singular value decomposition)**.** *Let $T : U \to V$ be a linear mapping between finite-dimensional vector spaces U and V. Then, there exist two orthonormal bases $\mathcal{B} = \{\mathbf{v}_1, \ldots, \mathbf{v}_n\} \subset V$ and $\mathcal{C} = \{\mathbf{u}_1, \ldots, \mathbf{u}_k\} \subset U$ such that the matrix of τ relative to $\mathcal{B}$ and $\mathcal{C}$ is diagonal:*

$$[T]_{\mathcal{B},\mathcal{C}} = \left[\begin{array}{ccc|c} \sigma_1 & & & 0 \\ & \ddots & & \vdots \\ & & \sigma_r & 0 \\ \hline 0 & \cdots & 0 & 0 \end{array}\right]$$

where $r = \operatorname{rank} T$ and $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0$.

If we reduce this case to be only between Euclidean spaces, i.e. $V = \mathbb{R}^n$ and $W = \mathbb{R}^k$, we can actually encode the linear map T in terms of a matrix $\mathbf{A}$, i.e. via $\mathbf{x} \mapsto \mathbf{A}\mathbf{x}$. Given change of bases $\underset{\mathcal{C} \leftarrow \boldsymbol{\varepsilon}_k}{\mathbf{P}}$ and $\underset{\boldsymbol{\varepsilon}_n \leftarrow \mathcal{B}}{\mathbf{P}}$ where $\boldsymbol{e}_i$ is the standard basis of $\mathbb{R}^i$, we have

$$[T]_{\mathcal{B},\mathcal{C}} = \underset{\mathcal{C} \leftarrow \boldsymbol{\varepsilon}_k}{\mathbf{P}} \underset{}{\mathbf{A}} \underset{\boldsymbol{\varepsilon}_n \leftarrow \mathcal{B}}{\mathbf{P}}$$

Multiplying by the inverses of the change-of-basis matrices, we have

$$\mathbf{A} = \underset{\boldsymbol{\varepsilon}_k \leftarrow \mathcal{C}}{\mathbf{P}} [T]_{\mathcal{B},\mathcal{C}} \underset{\boldsymbol{\varepsilon}_n \leftarrow \mathcal{B}}{\mathbf{P}}^{-1} = \begin{bmatrix}\mathbf{u}_1 \cdots \mathbf{u}_k\end{bmatrix} [T]_{\mathcal{B},\mathcal{C}} \begin{bmatrix}\mathbf{v}_1 \cdots \mathbf{v}_n\end{bmatrix}^{-1} = \begin{bmatrix}\mathbf{u}_1 \cdots \mathbf{u}_k\end{bmatrix} [\tau]_{\mathcal{B},\mathcal{C}} \begin{bmatrix}\mathbf{v}_1 \cdots \mathbf{v}_n\end{bmatrix}^{\intercal}$$

and if we let $\mathbf{U} = \begin{bmatrix}\mathbf{u}_1 \cdots \mathbf{u}_k\end{bmatrix}$, $\boldsymbol{\Sigma} = [T]_{\mathcal{B},\mathcal{C}}$, and $\mathbf{V} = \begin{bmatrix}\mathbf{v}_1 \cdots \mathbf{v}_n\end{bmatrix}$ (for simplicity purposes), we have

$$\mathbf{A} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^{\intercal}$$

hence the following equivalent statement of Theorem 1.1:

Theorem 1.2 (Singular value decomposition)**.** *Let $\mathbf{A} \in \mathcal{M}_{k \times n}(\mathbb{R})$. There exist two orthogonal matrices $\mathbf{U}$ and $\mathbf{V}$ such that*

$$\mathbf{A} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^{\intercal}$$

and Σ is a diagonal matrix of the form

$$\boldsymbol{\Sigma} = \left[\begin{array}{ccc|c} \sigma_1 & & & 0 \\ & \ddots & & \vdots \\ & & \sigma_r & 0 \\ \hline 0 & \cdots & 0 & 0 \end{array}\right]$$

where $r = \operatorname{rank} \mathbf{A}$ and $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0$.

In finding the matrices $\mathbf{U}$, $\boldsymbol{\Sigma}$, and $\mathbf{V}$ specified in Theorem 1.2, we go through the following algorithm:

1. The columns of $\mathbf{V}$, $\{\mathbf{v}_1, \ldots, \mathbf{v}_n\} \subset \mathbb{R}^n$, form an orthonormal eigenbasis of $\mathbf{A}^{\intercal}\mathbf{A}$.
2. Reorder $\{\mathbf{v}_1, \ldots, \mathbf{v}_n\}$ such that their corresponding eigenvalues are listed in equal or decreasing order. The diagonal values $\sigma_1, \ldots, \sigma_r$ of $\boldsymbol{\Sigma}$ are defined $\sigma_i = \sqrt{\lambda_i}$ and they should be listed in Σ in the same order as the eigenvectors are in $\mathbf{V}$.

3. Define the first r columns of $\mathbf{U}$ as $\{\mathbf{u}_1, \dots, \mathbf{u}_r\} = \left\{ \frac{1}{\sigma_1}\mathbf{A}\mathbf{v}_1, \dots, \frac{1}{\sigma_r}\mathbf{A}\mathbf{v}_r \right\} \subset \operatorname{col} \mathbf{A}$.

4. For the rest of the columns, find a basis for $(\operatorname{col} \mathbf{A})^\perp = \operatorname{null} \mathbf{A}^\intercal$. Perform Gram–Schmidt orthonormalization to develop a complete orthonormal basis $\{\mathbf{u}_1, \dots, \mathbf{u}_k\}$, from which we construct $\mathbf{U}$.

Despite its overwhelmingly complicated algorithm, the singular value decomposition (henceforth called SVD) can elicit some interesting results about the matrix $\mathbf{A}$ in question. First of all, by the end we have found orthonormal bases for the four fundamental subspaces: $\{\mathbf{u}_1, \dots, \mathbf{u}_r\} \subset \operatorname{col} \mathbf{A}$, $\{\mathbf{u}_{r+1}, \dots, \mathbf{u}_k\} \subset \operatorname{null} \mathbf{A}^\intercal$, $\{\mathbf{v}_1, \dots, \mathbf{v}_r\} \subset \operatorname{col} \mathbf{A}^\intercal$, and $\{\mathbf{v}_{r+1}, \dots, \mathbf{v}_n\} \subset \operatorname{null} \mathbf{A}$. In a way, these four sets of orthonormal bases essentially characterize the nature of the matrix (even though the SVD factorization may not be unique).

Second of all, an intuitive motivation is to visualize $\mathbf{A}$ as a linear transformation between orthonormal bases. What the SVD lists is an orthonormal basis, say $\{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3\}$, such that, when transformed by $\mathbf{A}$, it gives an *orthogonal* basis $\{\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3\}$. Thus we can express three equations: $\mathbf{A}\mathbf{v}_1 = \mathbf{y}_1$, $\mathbf{A}\mathbf{v}_2 = \mathbf{y}_2$, and $\mathbf{A}\mathbf{v}_3 = \mathbf{y}_3$. To normalize the basis of $\mathbf{y}_i$, we divide by their norms, which we call the singular value $\sigma_i = \|\mathbf{y}_i\|$, thus giving us a new orthonormal basis $\{\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3\}$. From the three equations we gain the system $\mathbf{A}\mathbf{v}_i = \sigma_i \mathbf{u}_i$, which we can automatically gain the matrix equation $\mathbf{A}\begin{bmatrix}\mathbf{v}_1 & \mathbf{v}_2 & \mathbf{v}_3\end{bmatrix} = \begin{bmatrix}\sigma_1\mathbf{u}_1 & \sigma_2\mathbf{u}_2 & \sigma_3\mathbf{u}_3\end{bmatrix}$, which then reformats as $\mathbf{A}\begin{bmatrix}\mathbf{v}_1 & \mathbf{v}_2 & \mathbf{v}_3\end{bmatrix} = \begin{bmatrix}\mathbf{u}_1 & \mathbf{u}_2 & \mathbf{u}_3\end{bmatrix}\operatorname{diag}(\sigma_1, \sigma_2, \sigma_3)$. The abbreviation of this that we conserve for the SVD statement is $\mathbf{A}\mathbf{V} = \mathbf{U}\mathbf{\Sigma}$.

Third of all, this gives us a way to **approximate** matrices while preserving their "general structure." Let's say we have a matrix $\mathbf{A} \in \mathcal{M}_{k\times n}(\mathbb{R})$ with rank r—then we can generate a **rank m approximation** (where $m \leq r$) which we denote $\mathbf{A}_m$. This matrix $\mathbf{A}_m$ would therefore have to be decomposed as $\mathbf{A}_m \coloneqq \mathbf{U}_m\mathbf{\Sigma}_m\mathbf{V}_m^\intercal$ where we take the first m columns of $\mathbf{U}$ ($\mathbf{U}_m$), the first m columns of $\mathbf{V}$ ($\mathbf{V}_m$), and the first square sub-block of $\mathbf{\Sigma}$ ($\mathbf{\Sigma}_m$).

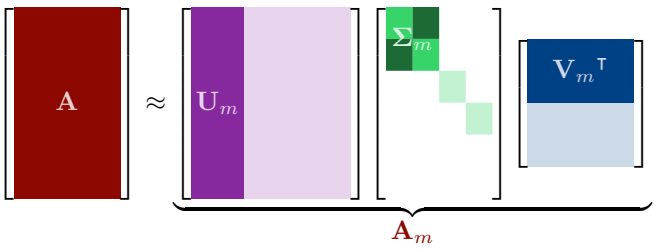

Figure 1: $\mathbf{A}_m$ is a rank m approximation of $\mathbf{A}$, which records the first m singular values coupled with their respective orthonormal bases. This is the mechanism behind SVD-based image compression.

Using this "rank m approximation," we can store a **selectable proportion of the energy** contained in the original matrix $\mathbf{A}$. Note that the first few singular values capture the majority of the aforementioned energy; as the approximation $\mathbf{A}_m$ increases in rank m, the singular values add decreasing amounts of energy. This is a consequence from the definition of the singular value decomposition which states that all singular values are positive and non-increasing, i.e. $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r \geq 0$.

The validity of the SVD in computing rank m approximation lies in the Eckart–Young–Mirsky theorem of 1936, which essentially states that the **best possible rank m approximation** of $\mathbf{A}$ is the rank m truncation of the SVD and none else:

Definition 1.1. *The **Frobenius norm** of a matrix $\mathbf{A} \in \mathcal{M}_{k\times n}(\mathbb{R})$ is the value $\|\mathbf{A}\|_F = \sqrt{\sum_{i,j} a_{ij}^2}$.*

Theorem 1.3 (Eckart–Young–Mirsky)**.** *For $\mathbf{A} \in \mathcal{M}_{k\times n}(\mathbb{R})$, we have* $\operatorname*{argmin}_{\tilde{\mathbf{X}}\text{ s.t. }\operatorname{rank}\mathbf{X}=m} \left\|\mathbf{X} - \tilde{\mathbf{X}}\right\|_F = \mathbf{A}_m$ *[1].*

2 Image compression

The method of taking a smaller rank approximation through SVD allows us to "compress" matrix-like objects, such as **raster images**, under the motivation of minimizing the amount of stored information, all the while preserving most of the quality in the image. In this instance, we use the MATLAB programming language to execute image compression.

The image to be compressed is a portrait of the author of this paper, `'Picardo.Robert.jpg'`, and it has dimensions 4032×3024. It was made grayscale through `rgb2gray` so to limit the entries of the image matrix in one dimension ($\mathbb{R}$).

```
clear all, close all, clc

ranks = [5 10 50];
fig1 = figure;

A = imread('Picardo.Robert.jpg');
X = double(rgb2gray(A));

subplot(2,2,1)
imshow(uint8(X)), axis off
title('Original');

set(gcf,'Position',[1400 100 1200 1600]);

subplotindex = 2;

% Singular value decomposition
[U, S, V] = svd(X,'econ');
for r = ranks
    Xapprox = U(:,1:r) * S(1:r,1:r) * V(:,1:r)'; % rank r approximation
    subplot(2,2,subplotindex);
    subplotindex = subplotindex + 1;
    imshow(uint8(Xapprox)), axis off
    label = ['$r = ', num2str(r), ', ' num2str(100 * r*sum(size(X))/numel(X)), '\%$ storage'];
    title(label, 'interpreter', 'latex')
end
```

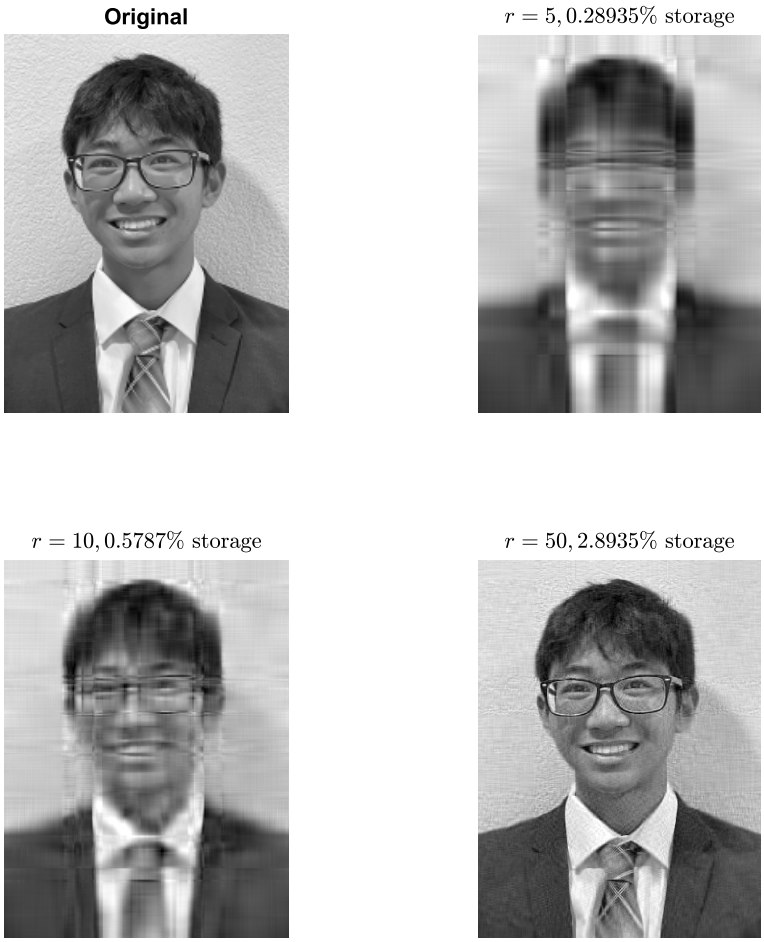

Figure 2: The result of the MATLAB code above. Aside from the original, each compressed image has rank r, which corresponds to the number of singular values within the approximation.

It is important to note the storage space occupied by the compressed image. Compared to the original image, which would hold $k \times n$ values in its array, the SVD would only need to occupy $m(1+k+n)$ values, for a rank

m approximation $\mathbf{A}_m$. This number comes from the fact that

$$\mathbf{A} \approx \mathbf{A}_m = \sum_{i=1}^{m} \sigma_i \mathbf{u}_i \mathbf{v}_i^\intercal$$

In Figure 2, we calculate the proportion of energy storage $\chi(\mathbf{A}_m)$ compared to the storage of the original image with the following equation:

$$\chi(\mathbf{A}_m) = \frac{m(1+k+n)}{kn}$$

given a rank m approximation $\mathbf{A}_m$ of $\mathbf{A} \in \mathcal{M}_{k\times n}(\mathbb{R})$. In determining the matrix's SVD factorization, it is especially important to keep track of the singular values in the process. For a rank r approximation of $\mathbf{A}$, the singular value decreases **somewhat hyperbolically** as r increases—in the minimal values of r, the singular values are relatively large to the "long tail" of small singular values, which can be interpreted to "fill in the smaller details" of the image.

```
fig2 = figure;

x = 1:size(X,2);
y = diag(S);
plot(x,y,'LineWidth',2)
set(gca, 'YScale', 'log')
set(gca, 'FontSize', 14)
xlim([-50 3024])
xlabel('$r$', 'interpreter', 'latex')
ylabel('Singular value $\sigma_r$', 'interpreter', 'latex')
```

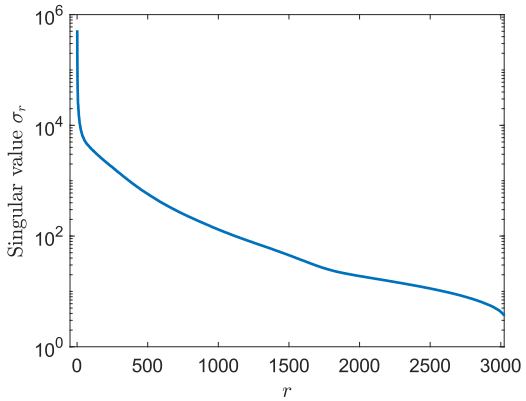

Figure 3: The rth singular value of the uint8 matrix corresponding to the image `'Picardo.Robert.jpg'`, plotted on a logarithmic scale along the σ_r-axis.

One can observe from Figure 3 that most of the **cumulative energy** from the original image is preserved within the SVD-truncations of the image with up to rank $r \approx 80$, where the singular values *start to decrease less* as r increases. This can serve as an eyeball estimate for a "sufficiently good" SVD-truncation, but there does exist a rigorous method for finding the **optimal truncation threshold**. We discuss this in the next section.

3 Optimal SVD-truncation

One of the bigger issues of SVD-based image compression is the tradeoff between data complexity and depiction accuracy when choosing an SVD-truncation of an image $\mathbf{X}$. As discussed above, a rule of thumb used to determine a best image compression would usually be to examine where there situates a "large gap" or "elbow," were the singular values to be plotted in nonincreasing order. However, a non-eyeball **optimal hard threshold of singular values**, which we denote τ_* and thus truncate all $\sigma_i > \tau_*$, can be determined through two cases, as found by Gavish and Donoho (2014) [2].

Suppose we have a *reference* matrix $\mathbf{X} \in \mathcal{M}_{m\times n}(\mathbb{R})$ such that it can be decomposed as such:

$$\mathbf{X} = \mathbf{X}_{\text{true}} + \gamma \mathbf{X}_{\text{noise}}$$

where $\mathbf{X}_{\text{true}} \in \mathcal{M}_{m \times n}(\mathbb{R})$ is some matrix containing the "true" information, and $\mathbf{X}_{\text{noise}}$ is a **noise matrix** with independent and identically distributed (i.e. "random") entries with mean 0.

If $\mathbf{X}$ is a square $n \times n$ matrix, then we set the optimal hard threshold to be

$$\tau_* = \frac{4}{\sqrt{3}} \gamma \sqrt{n}$$

where γ is the noise level (i.e. coefficient of the noise matrix).

If $\mathbf{X}$ was instead a non-square $m \times n$ matrix with $m \neq n$, then we instead set the optimal hard threshold to be

$$\tau_* = \lambda_*(\beta) \gamma \sqrt{n}$$

where $\beta = m/n$ and

$$\lambda_*(\beta) \coloneqq \sqrt{2(\beta+1) + \frac{8\beta}{(\beta+1) + \sqrt{\beta^2 + 14\beta + 1}}}$$

However, **if the noise level γ is unknown** (i.e. we are given only the matrix $\mathbf{X}$ and nothing else), then we will have to approximate the optical location of truncation τ_* by a slightly less robust value:

$$\hat{\tau}_* \approx 2.858 \, \sigma_{\text{med}}$$

where σ_{med} is the median singular value of $\mathbf{X}$, a square matrix. If $\mathbf{X}$ was not square, then our approximation obtains

$$\hat{\tau}_* = \omega(\beta) \, \sigma_{\text{med}}$$

where

$$\omega(\beta) \approx 0.56\beta^3 - 0.95\beta^2 + 1.82\beta + 1.43$$

($\omega(\beta)$ can be calculated algorithmically but not analytically; see Gavish and Donoho for more details) [2].

References

[1] C. Eckart and G. Young, "The approximation of one matrix by another of lower rank," *Psychometrika*, vol. 1, no. 3, pp. 211–218, Sep 1936. [Online]. Available: https://doi.org/10.1007/BF02288367

[2] M. Gavish and D. L. Donoho, "The optimal hard threshold for singular values is $4/\sqrt{3}$," *IEEE Transactions on Information Theory*, vol. 60, no. 8, pp. 5040–5053, Aug 2014.