Singular Value Decomposition

(COMS 4770/5770 Notes)

Yan-Bin Jia

Oct 3, 2024

1 Introduction

Now comes a highlight of linear algebra. Any real $m \times n$ matrix can be factored as

$$A = U\Sigma V^\top$$

where U is an $m \times m$ orthogonal matrix[1] whose columns are the eigenvectors of $AA^\top$, V is an $n \times n$ orthogonal matrix whose columns are the eigenvectors of $A^\top A$, and Σ is an $m \times n$ diagonal matrix of the form

$$\Sigma = \begin{pmatrix} \sigma_1 & & & & \\ & \ddots & & 0 & \\ & & \sigma_r & & \\ & & & 0 & \\ & 0 & & \ddots & \\ & & & & 0 \end{pmatrix}$$

with $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0$ and $r = \text{rank}(A)$. In the above, $\sigma_1, \ldots, \sigma_r$ are the square roots of the eigenvalues of $A^\top A$. They are called the *singular values* of A.

Our basic goal is to "solve" the system $A\boldsymbol{x} = \boldsymbol{b}$ for all matrices A and vectors $\boldsymbol{b}$. A second goal is to solve the system using a numerically stable algorithm. A third goal is to solve the system in a reasonably efficient manner. For instance, we do not want to compute A^{-1} using determinants.

Three situations arise regarding the basic goal:

(i) If A is square and invertible, we want to have the solution $\boldsymbol{x} = A^{-1}\boldsymbol{b}$.

(ii) If A is underconstrained, we want the entire set of solutions.

(iii) If A is overconstrained, we could simply give up. But this case arises a lot in practice, so instead we will ask for the least-squares solution. In other words, we want that $\bar{\boldsymbol{x}}$ which minimizes the error $\|A\bar{\boldsymbol{x}} - \boldsymbol{b}\|$. Geometrically, $A\bar{\boldsymbol{x}}$ is the point in the column space of A closest to $\boldsymbol{b}$.

[1] That is, $UU^\top = U^\top U = I$.

Gaussian elimination is reasonably efficient, but it is not numerically very stable. In particular, elimination does not deal with nearly singular matrices. The method is not designed for overconstrained systems. Even for underconstrained systems, the method requires extra work.

The poor numerical character of elimination can be seen in a couple ways. First, the elimination process assumes a non-singular matrix. But singularity, and rank in general, is a slippery concept. After all, the matrix A contains continuous, possibly noisy, entries. Yet, rank is a discrete integer. Strictly speaking, the two sets below are linearly independent vectors:

$$\left\{ \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \right\}, \qquad \left\{ \begin{pmatrix} 1.01 \\ 1.00 \\ 1.00 \end{pmatrix}, \begin{pmatrix} 1.00 \\ 1.01 \\ 1.00 \end{pmatrix}, \begin{pmatrix} 1.00 \\ 1.00 \\ 1.01 \end{pmatrix} \right\}.$$

Yet, the first set seems genuinely independent, while the second set seems "almost dependent".

Second, elimination-based methods work like LU decomposition, which represents the coefficient matrix A as a matrix product LDU, where L and U are respectively lower and upper diagonal and D is diagonal. One solves the system $A\boldsymbol{x} = \boldsymbol{b}$ by solving (via backsubstitution) $L\boldsymbol{y} = \boldsymbol{b}$ and $U\boldsymbol{x} = D^{-1}\boldsymbol{y}$. If A is nearly singular, then D will contain near-zero entries on its diagonal, and thus D^{-1} will contain large numbers. That is OK since in principle one needs the large numbers to obtain a true solution. The problem is if A contains noisy entries. Then the large numbers may be pure noise that dominates the true information. Furthermore, since L and U can be fairly arbitrary, they may distort or magnify that noise across other variables.

Singular value decomposition (SVD) is a powerful technique for dealing with sets of equations or matrices that are either singular or else numerically very close to singular. In many cases where Gaussian elimination and LU decomposition fail to give satisfactory results, SVD will not only diagnose the problem but also give you a useful numerical answer. It is also the method of choice for solving most linear least-squares problems.

2 SVD Close-up

An $n \times n$ symmetric matrix A has an eigen decomposition in the form of

$$A = S\Lambda S^{-1},$$

where Λ is a diagonal matrix with the eigenvalues δ_i of A on the diagonal and S contains the eigenvectors of A.

Why is the above decomposition appealing? The answer lies in the change of coordinates $\boldsymbol{y} = S^{-1}\boldsymbol{x}$. Instead of working with the system $A\boldsymbol{x} = \boldsymbol{b}$, we can work with the system $\Lambda\boldsymbol{y} = \boldsymbol{c}$, where $\boldsymbol{c} = S^{-1}\boldsymbol{b}$. Since Λ is diagonal, we are left with a trivial system

$$\delta_i y_i = c_i, \qquad i = 1, \ldots, n.$$

If this system has a solution, then another change of coordinates gives us $\boldsymbol{x}$, that is, $\boldsymbol{x} = S\boldsymbol{y}$.[2]

[2]There is no reason to believe that computing $S^{-1}\boldsymbol{b}$ is any easier than computing $A^{-1}\boldsymbol{b}$. However, in the ideal case, the eigenvectors of A are orthogonal. This is true, for instance, if $AA^\top = A^\top A$. In that case the columns of S are orthogonal and so we can take S to be orthogonal. But then $S^{-1} = S^\top$, and the problem of solving $A\boldsymbol{x} = \boldsymbol{b}$ becomes very simple.

Unfortunately, the decomposition $A = S\Lambda S^{-1}$ is not always possible. The condition for its existence is that A is $n \times n$ symmetric with n linearly independent eigenvectors. Even worse, what do we do if A is not square?

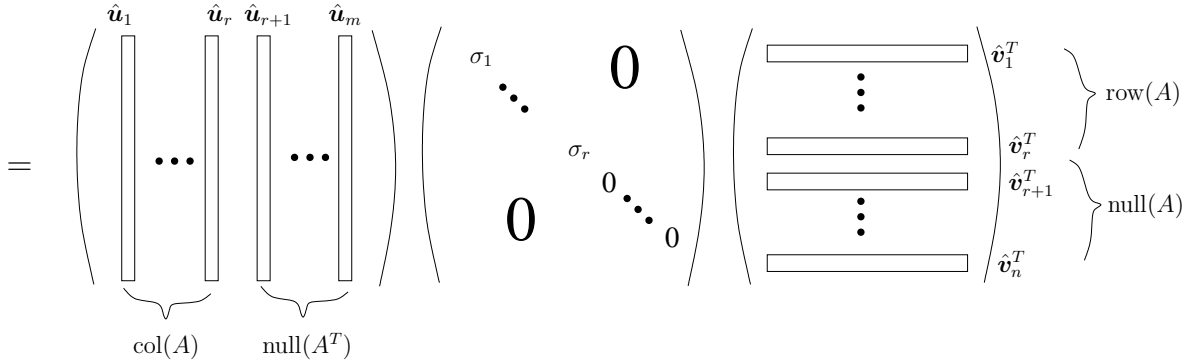
The answer is to work with $A^\top A$ and $AA^\top$, both of which are symmetric (and have n and m orthogonal eigenvectors, respectively). So we have the following decompositions:

$$\begin{aligned} A^\top A &= VDV^\top, \\ AA^\top &= UD'U^\top, \end{aligned}$$

where V is an $n \times n$ orthogonal matrix consisting of orthogonal unit eigenvectors $\hat{\boldsymbol{v}}_1, \hat{\boldsymbol{v}}_2, \ldots, \hat{\boldsymbol{v}}_n$ of $A^\top A$, D an $n \times n$ diagonal matrix with the corresponding eigenvalues of $A^\top A$ on the diagonal, U an $m \times m$ orthogonal matrix consisting of orthogonal unit eigenvectors $\hat{\boldsymbol{u}}_1, \hat{\boldsymbol{u}}_2, \ldots, \hat{\boldsymbol{u}}_m$ of $AA^\top$, and D' an $m \times m$ diagonal matrix with the corresponding eigenvalues of $AA^\top$ on the diagonal. It turns out that D and D' have the same non-zero diagonal entries except that the order might be different. The indices of the eigenvectors $\hat{\boldsymbol{u}}_i$ and $\hat{\boldsymbol{v}}_j$ can be permuted so that $\hat{\boldsymbol{u}}_k$ and $\hat{\boldsymbol{v}}_k$, $1 \le k \le r$, where r is the rank of A, always correspond to the same non-zero eigenvalue.

The SVD of A takes the following form:

$$\underset{m \times n}{A} = \underset{m \times m}{U} \quad \underset{m \times n}{\Sigma} \quad \underset{n \times n}{V^\top} \tag{1}$$



There are several facts about SVD:

(i) $\text{rank}(A) = \text{rank}(\Sigma) = r$.

(ii) The column space of A is spanned by the first r columns of U.

(iii) The null space of A is spanned by the last $n - r$ columns of V.

(iv) The row space of A is spanned by the first r columns of V.

(v) The null space of $A^\top$ is spanned by the last $m - r$ columns of U.

We can think of U and V as rotations and reflections and Σ as stretching matrix. Figure 1 illustrates the sequence of transformation under A on the unit vectors $\hat{\boldsymbol{v}}_1$ and $\hat{\boldsymbol{v}}_2$ (and all other vectors on the unit circle) in the case $m = n = 2$. Note that $V = (\hat{\boldsymbol{v}}_1 \hat{\boldsymbol{v}}_2)$. When multiplied by $V^\top$

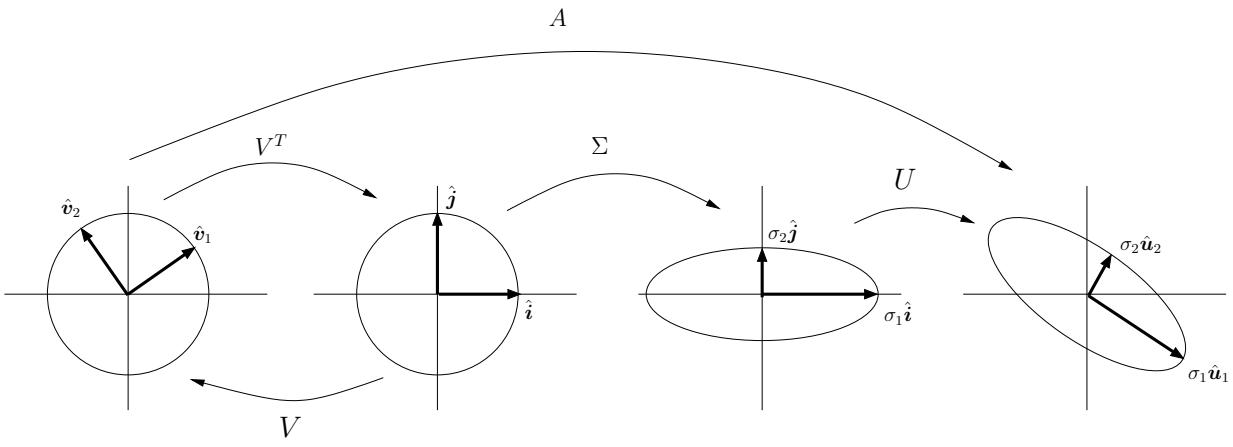


Figure 1: SVD illustrated as a transformation in three steps.

on the left, the two vectors undergo a rotation and become unit vectors $\hat{\boldsymbol{i}} = (1,0)^\top$ and $\hat{\boldsymbol{j}} = (0,1)^\top$. Then the matrix Σ stretches these two resulting vectors to $\sigma_1\hat{\boldsymbol{i}}$ and $\sigma_2\hat{\boldsymbol{j}}$, respectively. In the last step, the vectors undergo a final rotation due to U and become $\sigma_1\hat{\boldsymbol{u}}_1$ and $\sigma_2\hat{\boldsymbol{u}}_2$.

From (1) we also see that

$$A = \sigma_1\hat{\boldsymbol{u}}_1\hat{\boldsymbol{v}}_1^\top + \sigma_2\hat{\boldsymbol{u}}_2\hat{\boldsymbol{v}}_2^\top + \cdots + \sigma_r\hat{\boldsymbol{u}}_r\hat{\boldsymbol{v}}_r^\top. \tag{2}$$

We can swap σ_i with σ_j as long as we swap $\hat{\boldsymbol{u}}_i$ with $\hat{\boldsymbol{u}}_j$ and $\hat{\boldsymbol{v}}_i$ with $\hat{\boldsymbol{v}}_j$ at the same time. If $\sigma_i = \sigma_j$, then $\hat{\boldsymbol{u}}_i$ and $\hat{\boldsymbol{u}}_j$ can be swapped as long as $\hat{\boldsymbol{v}}_i$ and $\hat{\boldsymbol{v}}_j$ are also swapped. SVD is unique up to the permutations of $(\hat{\boldsymbol{u}}_i, \sigma_i, \hat{\boldsymbol{v}}_i)$ or of $(\hat{\boldsymbol{u}}_i, \hat{\boldsymbol{v}}_i)$ among those with equal σ_is. It is also unique up to the signs of $\hat{\boldsymbol{u}}_i$ and $\hat{\boldsymbol{v}}_i$, which have to change simultaneously.

Example 1. Find the singular value decomposition of

$$A = \begin{pmatrix} 2 & 2 \\ -1 & 1 \end{pmatrix}.$$

The eigenvalues of

$$A^\top A = \begin{pmatrix} 5 & 3 \\ 3 & 5 \end{pmatrix}$$

are 2 and 8 corresponding to unit eigenvectors

$$\hat{\boldsymbol{v}}_1 = \begin{pmatrix} -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix} \qquad \text{and} \qquad \hat{\boldsymbol{v}}_2 = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix},$$

respectively. Hence $\sigma_1 = \sqrt{2}$ and $\sigma_2 = \sqrt{8} = 2\sqrt{2}$. We have

$$\begin{aligned} A\hat{\boldsymbol{v}}_1 &= \sigma_1\hat{\boldsymbol{u}}_1\hat{\boldsymbol{v}}_1^\top\hat{\boldsymbol{v}}_1 = \sigma_1\hat{\boldsymbol{u}}_1 = \begin{pmatrix} 0 \\ \sqrt{2} \end{pmatrix}, \qquad \text{so} \quad \hat{\boldsymbol{u}}_1 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}; \\ A\hat{\boldsymbol{v}}_2 &= \sigma_2\hat{\boldsymbol{u}}_2\hat{\boldsymbol{v}}_2^\top\hat{\boldsymbol{v}}_2 = \sigma_1\hat{\boldsymbol{u}}_2 = \begin{pmatrix} 2\sqrt{2} \\ 0 \end{pmatrix}, \qquad \text{so} \quad \hat{\boldsymbol{u}}_2 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}. \end{aligned}$$

The SVD of A is therefore

$$A = U\Sigma V^\top = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} \sqrt{2} & 0 \\ 0 & 2\sqrt{2} \end{pmatrix} \begin{pmatrix} -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{pmatrix}.$$

Note that we can also compute the eigenvectors $\hat{\boldsymbol{u}}_1$ and $\hat{\boldsymbol{u}}_2$ directly from $AA^\top$.

3 Eigenvalues vs. Singular Values

There are several differences between eigenvalues and singular values.

(i) A matrix always has singular values defined, while its eigenvalues are defined only when it is a square matrix.

(ii) For a square matrix, its singular values are always non-negative while it may have negative or imagninary eigenvalues. For example, the matrix A in Example 1 has two imaginary eigenvalues $\frac{3\pm\sqrt{7}i}{2}$.

(iii) Eigenvectors of a square matrix A give the directions of invariant action. More specifically, when A acts on an eigenvector $\boldsymbol{a}$ corresponding to an eigenvalue λ, the result $A\boldsymbol{a} = \lambda\boldsymbol{a}$ simply shrinks or stretches it with its direction either unchanged or reversed. These invariant directions, however, do not necesssarily generate the extreme effects measured as by the relative change in the length of the vector, more specifically, by the ratio $\|A\boldsymbol{a}\|/\|\boldsymbol{a}\|$.

(iv) The unit eigenvectors $\hat{\boldsymbol{v}}_1, \hat{\boldsymbol{v}}_2, \ldots, \hat{\boldsymbol{v}}_n$ of $A^\top A$ in (1), along with unit eigenvectors that correspond to zero eigenvalues, if any, are precisely the directions that mark the extreme effects of the transformation under A. Suppose $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n$. They represent the (locally) extreme values of the function $\|A\boldsymbol{x}\|/\|\boldsymbol{x}\|$ over $\mathbb{R}^n$. The direction $\hat{\boldsymbol{v}}_1$ undergoes the most dramatic scaling of its length from 1 to $\|A\hat{\boldsymbol{v}}_1\| = \|\sigma_1\hat{\boldsymbol{u}}_1\| = \sigma_1$ under the transformation, while the direction $\hat{\boldsymbol{v}}_n$ undergoes the least. The unit sphere in $\mathbb{R}^n$ defined by $\hat{\boldsymbol{v}}_1, \hat{\boldsymbol{v}}_2, \ldots, \hat{\boldsymbol{v}}_n$ is transformed into an ellipsoid in $\mathbb{R}^r$ (see the rightmost ellipse in Figure 1). The ratio σ_1/σ_n of the maximum to minimum stretching is the *condition number* of A. It measures the sensitivity of the solution of $A\boldsymbol{x} = \boldsymbol{b}$ to a change of its right hand side. This number if infinite when the matrix is singular, i.e., $\sigma_n = 0$.

When A is symmetric, its singular values $\sigma_1, \sigma_2, \ldots, \sigma_n$ are the absolute values of its eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_n$, namely,

$$\sigma_1 = |\lambda_1|, \quad \sigma_2 = |\lambda_2|, \quad , \ldots, \quad \sigma_n = |\lambda_n|.$$

To see this, consider an eigenvalue λ of A that corresponds to the eigenvector $\boldsymbol{a}$. We have $A\boldsymbol{a} = \lambda\boldsymbol{a}$ and

$$\begin{aligned} A^\top A\boldsymbol{a} &= A^\top(\lambda\boldsymbol{a}) \\ &= \lambda A^\top\boldsymbol{a} \\ &= \lambda^2\boldsymbol{a}. \end{aligned}$$

Thus, λ^2 is an eigenvalue of $A^\top A$ with $\boldsymbol{a}$ staying as an eigenvector. We see that A and $A^\top A$ have exactly the same set of eigenvectors. The eigenvalues of $A^\top A$ are simply the squares of those of A. The singular values of A are the square roots of the eigenvalues of $A^\top A$, i.e., $\sqrt{\lambda^2} = |\lambda|$.

4 More Discussion

That $\text{rank}(A) = \text{rank}(\Sigma)$ tells us that we can determine the rank of A by counting the non-zero entries in Σ. In fact, we can do better. Recall that one of our complaints about Gaussian elimination was that it did not handle noise or nearly singular matrices well. SVD remedies this situation.

For example, suppose that an $n \times n$ matrix A is nearly singular. Indeed, perhaps A should be singular, but due to noisy data, it is not quite singular. This will show up in Σ, for instance, when all of the n diagonal entries in Σ are non-zero and some of the diagonal entries are almost zero.

More generally, an $m \times n$ matrix A may appear to have rank r, yet when we look at Σ we may find that some of the singular values are very close to zero. If there are l such values, then the "true" rank of A is probably $r - l$, and we would do well to modify Σ. Specifically, we should replace the l nearly zero singular values with zero.

Geometrically, the effect of this replacement is to reduce the column space of A and increase its null space. The point is that the column space is warped along directions for which $\sigma_i \approx 0$. In effect, solutions to $A\boldsymbol{x} = \boldsymbol{b}$ get pulled off to infinity (since $\sigma_i^{-1} \approx \infty$) along vectors that are almost in the null space. So, it is better to ignore the ith coordinate by zeroing σ_i.

Example 2. The matrix

$$A = \begin{pmatrix} 1.01 & 1.00 & 1.00 \\ 1.00 & 1.01 & 1.00 \\ 1.00 & 1.00 & 1.00 \end{pmatrix}$$

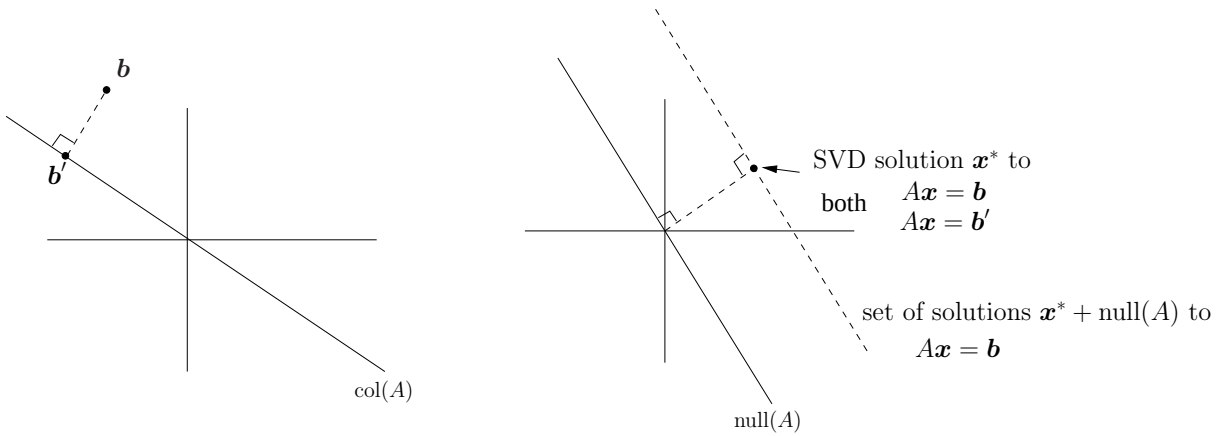
yields

$$\Sigma = \begin{pmatrix} 3.01 & 0 & 0 \\ 0 & 0.01 & 0 \\ 0 & 0 & 0.01 \end{pmatrix}.$$

Since 0.01 is significantly smaller than 3.01, we could treat it as zero and the rank A as one.

5 The SVD Solution

For the moment, let us suppose that A is an $n \times n$ square matrix. The following picture sketches the way in which SVD solves the system $A\boldsymbol{x} = \boldsymbol{b}$.



The system $A\boldsymbol{x} = \boldsymbol{b}$ has an affine set of solutions, given by $\boldsymbol{x}_0 + \text{null}(A)$, where $\boldsymbol{x}_0$ is any solution. It is easy to describe the null space $\text{null}(A)$ given the SVD decomposition, since it is just

the span of the last $n - r$ columns of V. Also note that $A\boldsymbol{x} = \boldsymbol{b}$ has no solution since $\boldsymbol{b}$ is not in the column space of A.

SVD solves $A\boldsymbol{x} = \boldsymbol{b}$ by determining that $\boldsymbol{x}$ which is the closest to the origin, i.e., which has the minimum norm. It first projects $\boldsymbol{b}$ onto the column space of A, obtaining $\boldsymbol{b}'$, and then solves $A\boldsymbol{x} = \boldsymbol{b}'$. Essentially, SVD obtains the *least-squares* solution.

Given the diagonal matrix Σ in the SVD of A, denote by $\Sigma^\dagger$ the diagonal matrix whose entries are of the form:

$$\left(\Sigma^\dagger\right)_{ij} = \begin{cases} \sigma_i^{-1} & \text{if } 1 \le i = j \le r; \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

Thus, the product matrix

$$\Sigma \cdot \Sigma^\dagger = \Sigma^\dagger \cdot \Sigma = \begin{pmatrix} 1 & & & & & \\ & \ddots & & & & \\ & & 1 & & & \\ & & & 0 & & \\ & & & & \ddots & \\ & & & & & 0 \end{pmatrix}$$

has r 1's on its diagonal. If Σ is invertible, then $\Sigma^\dagger = \Sigma^{-1}$.

Example 3. If

$$\Sigma = \begin{pmatrix} 2 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

then

$$\Sigma^\dagger = \begin{pmatrix} \frac{1}{2} & 0 & 0 \\ 0 & \frac{1}{5} & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

So how do we solve $A\boldsymbol{x} = \boldsymbol{b}$? We first compute the SVD decomposition $A = U\Sigma V^\top$. Then we use the solution

$$\bar{\boldsymbol{x}} = A^\dagger \boldsymbol{b}. \tag{4}$$

where

$$\begin{aligned} A^\dagger &= V\Sigma^\dagger U^\top & (5) \\ &= \sigma_1^{-1}\hat{\boldsymbol{v}}_1\hat{\boldsymbol{u}}_1^\top + \sigma_2^{-1}\hat{\boldsymbol{v}}_2\hat{\boldsymbol{u}}_2^\top + \cdots + \sigma_r^{-1}\hat{\boldsymbol{v}}_r\hat{\boldsymbol{u}}_r^\top & (6) \end{aligned}$$

is called the Moore-Penrose *pseudoinverse* of A. Compare equations (2) and (6) for similarities and differences.

The solution (4) and pseudoinverse (5) carry over when A is $m \times n$ with $\Sigma^\dagger$ still defined in (3). There are three possible cases below:

(i) If A is invertible, then $\bar{\boldsymbol{x}}$ is the unique solution to $A\boldsymbol{x} = \boldsymbol{b}$. Here, $A^\dagger = A^{-1}$.

(ii) If A is singular and $\boldsymbol{b}$ is in the column space of A, then $\bar{\boldsymbol{x}}$ is the solution closest to the origin. The set of solutions is $\boldsymbol{x} + \text{null}(A)$, where $\text{null}(A)$ is spanned by the last $n - r$ columns of V.

(iii) If A is singular and $\boldsymbol{b}$ is not in the column space of A, then $\bar{\boldsymbol{x}}$ is the least-squares solution.

In general there will not be any zero singular values. However, if A has column degeneracies there may be near-zero singular values. It may be useful to zero these out, to remove noise, as in the case of the matrix in Example 2. (The implication is that the overdetermined set is not quite as overdetermined as it seemed, that is, the null space is not trivial.)

6 The Moore-Penrose Pseudoinverse

The pseudoinverse $A^\dagger$ is the unique $n \times m$ matrix M that satisfies the following four conditions: i) $AMA = A$, ii) $MAM = A$, iii) $(AM)^\top = AM$, and iv) $(MA)^\top = MA$. The pseudoinverse has the following properties

$$\begin{aligned} A^\dagger \hat{\boldsymbol{u}}_i &= \begin{cases} \sigma_i^{-1} \hat{\boldsymbol{v}}_i & \text{if } i \le r, \\ 0 & \text{if } i > r; \end{cases} \\ \left(A^\dagger\right)^\top \hat{\boldsymbol{v}}_i &= \begin{cases} \sigma_i^{-1} \hat{\boldsymbol{u}}_i & \text{if } i \le r, \\ 0 & \text{if } i > r. \end{cases} \end{aligned}$$

Namely, the vectors $\hat{\boldsymbol{u}}_1, \ldots, \hat{\boldsymbol{u}}_r$ in the column space of A go back to the row space. The other vectors $\hat{\boldsymbol{u}}_{r+1}, \ldots, \hat{\boldsymbol{u}}_m$ are in the null space of $A^\top$, and $A^\dagger$ sends them to zero. When we know what happens to each basis vector $\hat{\boldsymbol{u}}_i$, we know $A^\dagger$ because $\hat{\boldsymbol{v}}_i$ and σ_i will be determined.

EXAMPLE 4. The pseudoinverse of $A = \begin{pmatrix} 2 & 2 \\ -1 & 1 \end{pmatrix}$ from Example 1 is $A^\dagger = A^{-1}$, because A is invertible. And we have

$$A^\dagger = A^{-1} = V \Sigma^{-1} U^\top = \begin{pmatrix} \frac{1}{4} & -\frac{1}{2} \\ \frac{1}{4} & \frac{1}{2} \end{pmatrix}.$$

Proposition 1 *The operations of transpose and pseudoinverse are commutative; that is, for every matrix A, the following holds:*

$$\left(A^\top\right)^\dagger = \left(A^\dagger\right)^\top.$$

Proof The matrix A of dimension, say, $m \times n$, has the SVD $A = U \Sigma V^\top$. First, the pseudoinverse of Σ, with dimension $n \times m$, is $\Sigma^\dagger$; and that of $\Sigma^\top$, with dimension $m \times n$, has the same non-zero diagonal entries $\sigma_1^{-1}, \sigma_2^{-1}, \ldots, \sigma_r^{-1}$. Hence, it trivially follows that

$$\left(\Sigma^\dagger\right)^\top = \left(\Sigma^\top\right)^\dagger.$$

Next, we have

$$\begin{aligned} \left(A^\top\right)^\dagger &= \left(V \Sigma^\top U^\top\right)^\dagger \\ &= U \left(\Sigma^\top\right)^\dagger V^\top \\ &= U \left(\Sigma^\dagger\right)^\top V^\top \\ &= \left(V \Sigma^\dagger U^\top\right)^\top \\ &= \left(A^\dagger\right)^\top. \end{aligned}$$

□

Lemma 2 *If the columns of an $m \times n$ matrix A are linearly independent, then the $n \times n$ matrix $A^\top A$ is non-singular. Similarly, if the rows of A are linearly independent, then the $m \times m$ matrix $AA^\top$ is non-singular.*

Proof Suppose A has linearly independent columns. It follows that

$$\begin{aligned}(A^\top A)\boldsymbol{x} = \mathbf{0} &\implies \boldsymbol{x}^\top A^\top A\boldsymbol{x} = \mathbf{0} \\ &\implies (A\boldsymbol{x})^\top (A\boldsymbol{x}) = 0 \\ &\implies \|A\boldsymbol{x}\| = 0 \\ &\implies A\boldsymbol{x} = \mathbf{0} \\ &\implies \boldsymbol{x} = \mathbf{0},\end{aligned}$$

where the last step follows from that the columns of A are linearly independent. Thus, the null space of $A^\top A$ contains only $\mathbf{0}$. The matrix is therefore non-singular.

Now suppose that A has linearly independent rows. Equivalently, $A^\top$ has linearly independent columns. Applying what we have just shown above, $(A^\top)^\top A^\top = AA^\top$ is non-singular. □

Theorem 3 *The following holds for the pseudoinverse of an $m \times n$ matrix A as defined in (5):*

$$A^\dagger = \begin{cases} (A^\top A)^{-1} A^\top & \text{if } \operatorname{rank}(A) = n; \\ A^\top (AA^\top)^{-1} & \text{if } \operatorname{rank}(A) = m. \end{cases} \tag{7}$$

Proof Consider the first situation where $\operatorname{rank}(A) = n$. we make use of the SVD of A:

$$A^\top A = V\Sigma^\top U^\top U\Sigma V^\top = V\Sigma^\top \Sigma V^\top,$$

where

$$\Sigma^\top \Sigma = \begin{pmatrix} \sigma_1^2 & & & \\ & \sigma_2^2 & & \\ & & \ddots & \\ & & & \sigma_n^2 \end{pmatrix}.$$

By Lemma 2, $A^\top A$ is non-singular. Subsequently,

$$\begin{aligned}(A^\top A)^{-1} A^\top &= V \left(\Sigma^\top \Sigma\right)^{-1} V^\top V \Sigma^\top U^\top \\ &= V \begin{pmatrix} 1/\sigma_1^2 & & \\ & \ddots & \\ & & 1/\sigma_n^2 \end{pmatrix} \begin{pmatrix} \sigma_1 & & & \\ & \ddots & & \mathbf{0} \\ & & \sigma_n & \end{pmatrix} U^\top \\ &= V \begin{pmatrix} 1/\sigma_1 & & & \\ & \ddots & & \mathbf{0} \\ & & 1/\sigma_n & \end{pmatrix} U^\top \\ &= V\Sigma^\dagger U^\top \\ &= A^\dagger.\end{aligned}$$

Consider the second situation where $\text{rank}(A) = m \leq n$. The product matrix $AA^\top$ is non-singular by Lemma 2 (if we substitute $A^\top$ for A). It is not difficult to carry out a sequence of steps similar to the above, establishing $A^\top(AA^\top)^{-1} = A^\dagger$. □

When $\text{rank}(A) = n$ (hence $m \geq n$), $A^\dagger = (A^\top A)^{-1}A^\top$ is called the *left inverse* of A since left multiplying A by $A^\dagger$ yields the identy matrix I_n. In this case, $A\boldsymbol{x} = \boldsymbol{b}$ is usually solved as follows. First, left multiply its both sides by $A^\top$ to obtain $A^\top A\boldsymbol{x} = A^\top\boldsymbol{b}$. Then, left multiply both sides of the resulting equation by the inverse of $A^\top A$:

$$\boldsymbol{x} = (A^\top A)^{-1}A^\top\boldsymbol{b}, \tag{8}$$

which, by Theorem 3, is essentially the SVD solution $\boldsymbol{x} = A^\dagger\boldsymbol{x}$ given in (4). This solution, however, is not exact unless the rank of the composite matrix (A, b) is also n.

When $\text{rank}(A) = m$ (hence $m \leq n$), $A^\dagger = A^\top(AA^\top)^{-1}$ is called the *right inverse* of A since a right multiplying A with $A^\dagger$ yields the identity matrix I_m. The SVD solution $\boldsymbol{x} = A^\dagger\boldsymbol{b}$ is an exact one but there exists a subspace of solutions unless $m = n$.

7 Optimal Approximation by a Matrix of Lower Rank

Almost all applications of the SVD stem from its use in the approximation of a matrix by a matrix of lower rank, with optimality established by the Eckart-Young-Mirsky theorem [4, 14]. Before presenting the theorem, let us go over matrix norms. The norm $\|\cdot\|$ of $m \times n$ matrices is a function such that, for any such matrix A, (i) $\|A\| \geq 0$ with the equality holding at $A = 0$ only, (ii) $\|cA\| = |c|\|A\|$ for any constant c, and (iii) $\|A + B\| \leq \|A\| + \|B\|$ holds given any other $m \times n$ matrix B. There two most frequently used matrix norms. One is the *Frobenius norm* defined as

$$\|A\|_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n} a_{ij}^2}, \tag{9}$$

and the other is the *p-norm* defined as

$$\|A\|_p = \max_{\boldsymbol{v}\neq\boldsymbol{0}} \frac{\|A\boldsymbol{v}\|_p}{\|\boldsymbol{x}\|_p}, \tag{10}$$

where $\|\boldsymbol{x}\|_p$ is the p-norm of the vector $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)^\top$ given as

$$\|\boldsymbol{x}\|_p = \left(|x_1|^p + |x_2|^p + \cdots |x_n|^p\right)^{1/p}.$$

Let r be the rank of A and $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r$ its singular values. Utilizing the decomposition (2) of A, we define

$$A_k = \sigma_1\hat{\boldsymbol{u}}_1\hat{\boldsymbol{v}}_1^\top + \cdots + \sigma_k\hat{\boldsymbol{u}}_k\hat{\boldsymbol{v}}_k^\top, \qquad 1 \leq k \leq r. \tag{11}$$

Theorem 4 (Eckart-Young-Mirsky) *The following holds under both the Frobenius norm $\|\cdot\|_F$ and the 2-norm $\|\cdot\|_2$:*

$$\|A - A_k\| \leq \|A - B\|$$

for any $m \times n$ matrix B of rank k. Furthermore,

$$\|A - A_k\| = \begin{cases} \left(\sum_{i=k+1}^{r}\sigma_i^2\right)^{1/2}, & \textit{under the Frobenius norm;} \\ \sigma_{k+1}, & \textit{under the 2-norm.} \end{cases}$$

Intuitively, the approximation by A_k retains the k most significant features of A. In the case that $\sigma_k \gg \sigma_{k+1}$ and $k \ll r$, such approximation reduces storage and computation significantly while keeping most of the information contained in A. This reduction in dimensionality achieves data compression. The principal component analysis (PCA) [8] is essentially based on the SVD, with the matrix A describing the covariances among data. Then it finds orthogonal axes along which data variances are (sequentially) maximized.

8 Applications

The SVD has been widely applied in image and signal processing. Image compression is achieved using equation (11) as a truncated SVD transformation to offer significant savings in storage [2, 21]. Image denoising and noise filtering [9] follow the same spirit, with the SVD applied to split an original image into a dominant subspace representing the signal, e.g., the image data, and a subdominant subspace representing the noise. Since smooth images have rapid decaying of singular values while noisy images have slow decaying of such values, the slope of singular values can be used as roughness measure [17]. In image forensics, SVD-based watermarking techniques employ either the largest or smallest singular values to embed the watermark components, leveraging their stability in specifying the luminance [11].

In robotics, the SVD has been used for kinematic and dynamic analysis, redundancy control, and dexterity optimization of manipulators [12]. In dynamics, it is applied to obtain periodic solutions to dissipative systems and Hamiltonian systems [13]. The SVD serves as both analysis and design tools to provide realistic answers to many practical questions that must be addressed in industrial processes [15]. In biology, gene expression data may undergo an SVD analysis to filter out noise and get classified into groups of similar traits such as regulation and phenotype [1].

In natural language processing, the SVD is implemented as part of a method called latent semantic analysis [3] to effectively discover semantic relationships within large texts and identify hidden patterns. Recommendation systems [10] uses the SVD to decompose a matrix representing user ratings on some features for prediction of unrated features.

A The SVD Algorithm

Here we only take a brief look at how the SVD algorithm actually works. For more details we refer to [6] and [7]. It is useful to establish a contrast with Gaussian elimination, which reduces a matrix A by a series of row operations that zero out portions of columns of A. Row operations imply pre-multiplying the matrix A. They are all collected together in the matrix L^{-1} where $A = LDU$.

In contrast, SVD zeros out portions of both rows and columns. Thus, whereas Gaussian elimination only reduces A using pre-multiplication, SVD uses both pre- and post-multiplication. As a result, SVD can at each stage rely on orthogonal matrices to perform its reductions on A. By using orthogonal matrices, SVD reduces the risk of magnifying noise and errors. The pre-multiplication matrices are gathered together in the matrix $U^\top$, while the post-multiplication matrices are gathered together in the matrix V.

There are two phases to the SVD decomposition algorithm:

(i) SVD reduces A to bidiagonal form using a series of orthogonal transformations. This phase is deterministic and has a running time that depends only on the size of the matrix A.

(ii) SVD removes the superdiagonal elements from the bidiagonal matrix using orthogonal transformation. This phase is iterative but converges quickly.

Let us take a slightly closer look at the first phase. Step 1 in this phase creates two orthogonal matrices U_1 and V_1 such that

$$U_1 A = \begin{pmatrix} a'_{11} & a'_{12} & \cdots & a'_{1n} \\ 0 & & & \\ \vdots & & B' & \\ 0 & & & \end{pmatrix},$$

$$U_1 A V_1 = \begin{pmatrix} a''_{11} & a''_{12} & 0 & \cdots & 0 \\ 0 & & & & \\ \vdots & & B'' & & \\ 0 & & & & \end{pmatrix}.$$

If A is $m \times n$ then B'' is $(m-1) \times (n-1)$. The next step of this phase recursively works on B'', and so forth, until orthogonal matrices $U_1, \ldots, U_{n-1}, V_1, \ldots, V_{n-2}$ are produced such that $U_{n-1} \cdots U_1 A V_1 \cdots V_{n-2}$ is bidiagonal (assume $m \geq n$).

In both phases, the orthogonal transformation are constructed from Householder matrices. For practitioners, free online libraries such as [22, 23] offer SVD source code in C++.

B Projection onto the Null Space

Using the pseudoinverse $A^\dagger$ we can project a n-vector onto the null space of A. This can be quite useful, for example, in the control of a robot during a task where its end-effector is moving on some surface while exerting force on the surface to maintain their contact, a situation where it beocomes possible to apply separate control policies on surface tracking and force regulation [20]. Let us start with the product of the pseudoinverse with matrix:

$$\begin{aligned} A^\dagger A &= V \Sigma^\dagger U^\top U \Sigma V^\top \\ &= V \Sigma^\dagger \Sigma V^\top \\ &= V \begin{pmatrix} 1 & & & & & \\ & \ddots & & & & \\ & & 1 & & & \\ & & & 0 & & \\ & & & & \ddots & \\ & & & & & 0 \end{pmatrix} V^\top, \end{aligned}$$

where the diagonal matrix in the last equation has r entries of 1. The symmetry of $A^\dagger A$ follows from that $\Sigma^\dagger \Sigma = \Sigma^\top \left(\Sigma^\dagger\right)^\top$. In the case that A has independent columns, the diagonal matrix becomes the identity matrix I_n and $A^\dagger A = V V^\top$.

Next, we define a *projection matrix* as follows:

$$P = I_n - A^\dagger A. \tag{12}$$

Clearly, P is also symmetric. Consider an n-vector $\boldsymbol{w}$. If $\boldsymbol{w} \in \text{null}(A)$, we easily see that

$$\begin{aligned} P\boldsymbol{w} &= (I_n - A^\dagger A)\boldsymbol{w} \\ &= \boldsymbol{w} - A^\dagger(A\boldsymbol{w}) \\ &= \boldsymbol{w}. \end{aligned}$$

If $\boldsymbol{w} \in \text{row}(A)$, then $\boldsymbol{w} = \lambda_1\hat{\boldsymbol{v}}_1 + \cdots + \lambda_r\hat{\boldsymbol{v}}_r$ for some $\lambda_1, \ldots, \lambda_r$. We have

$$\begin{aligned} P\boldsymbol{w} &= I_n\boldsymbol{w} - V\Sigma^\dagger\Sigma V^\top\boldsymbol{w} \\ &= \boldsymbol{w} - (\hat{\boldsymbol{v}}_1, \ldots, \hat{\boldsymbol{v}}_r, \hat{\boldsymbol{v}}_{r+1}, \ldots, \hat{\boldsymbol{v}}_n) \begin{pmatrix} 1 & & & & & & \\ & \ddots & & & & & \\ & & 1 & & & & \\ & & & 0 & & & \\ & & & & \ddots & & \\ & & & & & 0 & \end{pmatrix} \begin{pmatrix} \hat{\boldsymbol{v}}_1^\top \\ \vdots \\ \hat{\boldsymbol{v}}_r^\top \\ \hat{\boldsymbol{v}}_{r+1}^\top \\ \vdots \\ \hat{\boldsymbol{v}}_n^\top \end{pmatrix} \boldsymbol{w} \\ &= \boldsymbol{w} - (\hat{\boldsymbol{v}}_1, \ldots, \hat{\boldsymbol{v}}_r, \boldsymbol{0}, \ldots, \boldsymbol{0}) \begin{pmatrix} 1 & & & & & & \\ & \ddots & & & & & \\ & & 1 & & & & \\ & & & 0 & & & \\ & & & & \ddots & & \\ & & & & & 0 & \end{pmatrix} \begin{pmatrix} \hat{\boldsymbol{v}}_1^\top\boldsymbol{w} \\ \vdots \\ \hat{\boldsymbol{v}}_r^\top\boldsymbol{w} \\ 0 \\ \vdots \\ 0 \end{pmatrix} \\ &= \boldsymbol{w} - (\hat{\boldsymbol{v}}_1, \ldots, \hat{\boldsymbol{v}}_r) \begin{pmatrix} \hat{\boldsymbol{v}}_1^\top\boldsymbol{w} \\ \vdots \\ \hat{\boldsymbol{v}}_r^\top\boldsymbol{w} \end{pmatrix} \\ &= \boldsymbol{w} - \boldsymbol{w} \\ &= \boldsymbol{0}. \end{aligned}$$

In the general case, $\boldsymbol{w}$ is decomposed into $\boldsymbol{w}_n \in \text{null}(A)$ and $\boldsymbol{w}_r \in \text{row}(A)$. Hence we have

$$P\boldsymbol{w} = P\boldsymbol{w}_n + P\boldsymbol{w}_r = \boldsymbol{w}_n;$$

in other words, the matrix P projects away the vector's component in the row space while keeps its component in the null space. Finally, since the columns of $A^\top$ form the row space of A. We easily arrive at

$$PA^\top = \boldsymbol{0}.$$

Projection matrices with the same dimensions are commutative. To see this, let A_1 and A_2 be two matrices that have dimensions $m_1 \times n$ and $m_2 \times n$ and ranks r_1 and r_2, respectively. The intersection $\text{null}(A_1) \cap \text{null}(A_2)$ of their null spaces has the rank k. Consider n orthonormal vectors $\hat{\boldsymbol{v}}_1, \hat{\boldsymbol{v}}_2, \ldots, \hat{\boldsymbol{v}}_n$ such that $\text{null}(A_1) \cap \text{null}(A_2)$ is spanned by $\hat{\boldsymbol{v}}_{n-k+1}, \ldots, \hat{\boldsymbol{v}}_n$. Any vector $\boldsymbol{w} \in \mathbb{R}^n$ can be written as

$$\boldsymbol{w} = a_1\hat{\boldsymbol{v}}_1 + \cdots + a_{n-k}\hat{\boldsymbol{v}}_{n-k} + a_{n-k+1}\hat{\boldsymbol{v}}_{n-k+1} + \cdots + a_n\hat{\boldsymbol{v}}_n,$$

for some $a_1, a_2, \ldots, a_n \in \mathbb{R}$. Clearly, for $1 \le j \le n-k$,

$$0 = P_1P_2\hat{\boldsymbol{v}}_j = P_2P_1\hat{\boldsymbol{v}}_j,$$

since either $\hat{\boldsymbol{v}}_j$ must be in the row space of at least one of A_1 and A_2; and for $n-k+1 \leq j \leq n$,

$$\hat{\boldsymbol{v}}_j = P_1 P_2 \hat{\boldsymbol{v}}_j = P_2 P_1 \hat{\boldsymbol{v}}_j.$$

Hence, $P_1 P_2 \boldsymbol{w} = P_2 P_1 \boldsymbol{w}$. Since $\boldsymbol{w}$ is arbitrary, we infer that $P_1 P_2 = P_2 P_1$.

To generalize, for matrices A_i, $1 \leq i \leq l$, with dimensions $m_i \times n$, respectively, the product of their projection matrices, in an arbitrary order, always projects a vector in $\mathbb{R}^n$ onto their common null space $\cap_{i=1}^{l} \text{null}(A_i)$. The product matrix has zero entries only if the common null space becomes empty.

References

[1] O. Alter, P. O. Brown, and D. Botstein. Singular value decomposition for genome-wide expression data processing and modeling. *Proceedings of the National Academy of Sciences*, 97(18):10101–10106, 2000.

[2] H. Andrews and C. Patterson Singular Value Decomposition (SVD) Image Coding. *IEEE Transactions on Communications*, 24(4):425-432, 1976.

[3] S. Deerwester, S. T. Dumals, G. W. Furnas, T. K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407, 1990.

[4] C. Eckart and G. Young. The approximation of one matrix by another of lower rank, *Psychometrika*, 1:211–218, 1936.

[5] M. Erdmann. Lecture notes for *16-811 Mathematical Fundamentals for Robotics*. The Robotics Institute, Carnegie Mellon University, 1998.

[6] G. E. Forsythe, M. A. Malcolm, and C. B. Moler. *Computer Methods for Mathematical Computations*. Prentice-Hall, 1977.

[7] G. H. Golub and C. F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, 3rd edition, 1996.

[8] I. T. Jolliffe. *Principal Component Analysis*, Springer, 2nd edition, 2002.

[9] K. Konstantinides, B. Natarajan, and G. S. Yovanof. Noise estimation and filtering using block-based singular value decomposition. *IEEE Transactions on Image Process*, 6:479–483, 1997.

[10] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.

[11] R. Liu and T. Tan. A SVD-based watermarking scheme for protecting rightful ownership. *IEEE Transactions on Multimedia*, 4(1):121–128, 2002.

[12] A. A. Maciejewski and C. A. Klein. The singular value decomposition: computation and applications to robotics. *The International Journal of Robotics Research*, 8(6):63–79, 1989.

[13] J. P. Meijaard. Applications of the singular value decomposition in dynamics. *Computer Methods in Applied Mechanics and Engineering*, 103:151-173, 1993.

[14] L. Mirsky. Symmetric gauge functions and unitarily invariant norms, *The Quaterly Journal of Mathematics*, 11(1):50-59, 1960.

[15] C. Moore. Application of singular value decomposition to the design, analysis, and control of industrial processes. In *Proceedings of the American Control Conference*, pp. 643–650, 1986.

[16] W. H. Press, *et al. Numerical Recipes in C++: The Art of Scientific Computing*. Cambridge University Press, 2nd edition, 2002.

[17] R. A. Sadek. SVD based image processing applications: State of the art, contributions, and research challeenges. *International Journal of Advanced Computer Science and Applications*, 3(7):26–34, 2012.

[18] G. Strang. *Introduction to Linear Algebra*. Wellesley-Cambridge Press, 1993.

[19] Advait Tilak. Applications of Singular Value Decomposition (SVD), Essay for *COMS 5770 Foundations of Robotics and Computer Vision*. Iowa State University, Fall 2024.

[20] L. Villani and J. De Shutter. Force control. In *Handbook of Robotics, Part A*, B. Siciliano and O. Khatib (eds.), pp. 195–219, Springer, 2008.

[21] J. F. Yang and C. L. Lu. Combined techniques of singular value decomposition and vector quantization for image coding. *IEEE Transactions on Image Process*, 6(8): 1141–1146, 1995.

[22] Boost C++ Libraries. `http://www.boost.org/`.

[23] LIT-lib. `http://ltilib.sourceforge.net/doc/homepage/index.shtml`.