8. THE SINGULAR VALUE DECOMPOSITION

CMDA 3606; MARK EMBREE

VERSION OF 19 FEBRUARY 2017

The singular value decomposition (SVD) is among the most important and widely applicable matrix factorizations. It provides a natural way to untangle a matrix into its four fundamental subspaces, and reveals the relative importance of each direction within those subspaces. Thus the singular value decomposition is a vital tool for analyzing data, and it provides a slick way to understand (and prove) many fundamental results in matrix theory. It is the perfect tool for solving least squares problems, and provides the best way to approximate a matrix with one of lower rank. These notes construct the SVD in various forms, then describe a few of its most compelling applications.

8.1 Helpful facts about symmetric matrices

To derive the singular value decomposition of a general (rectangular) matrix $A \in \mathbb{C}^{m \times n}$, we shall rely on several special properties of the square, symmetric matrix $A^* A$. For this reason we first recall some fundamental results from the theory of symmetric matrices.

We shall use the term “symmetric” to mean a matrix where $A^* = A$. Often such matrices are instead called “Hermitian” or “self-adjoint,” and the term “symmetric” is reserved for matrices with $A^T = A$. If the entries of $A$ are all real, there is no distinction.

Theorem 8.1 (Spectral Theorem) Suppose $H \in \mathbb{C}^{n \times n}$ is symmetric. Then there exist $n$ (not necessarily distinct) eigenvalues $\lambda_1, \ldots, \lambda_n$ and corresponding unit-length eigenvectors $v_1, \ldots, v_n$ such that

$$Hv_j = \lambda_j v_j.$$  

The eigenvectors form an orthonormal basis for $\mathbb{C}^n$:

$$\mathbb{C}^n = \text{span}\{v_1, \ldots, v_n\}$$

and $v_j^* v_k = 0$ when $j \neq k$, and $v_j^* v_j = \|v_j\|^2 = 1$.

As a consequence of the Spectral Theorem, we can write any symmetric matrix $H \in \mathbb{C}^{n \times n}$ in the form

$$H = \sum_{j=1}^{n} \lambda_j v_j v_j^*.$$  \hspace{1cm} (8.1)
This equation expresses $H$ as the sum of the special rank-1 matrices $\lambda_j v_j v_j^*$. The singular value decomposition will provide a similar way to tease apart a rectangular matrix.

**Theorem 8.2** All eigenvalues of a symmetric matrix are real.

**Proof.** Let $(\lambda_j, v_j)$ be an arbitrary eigenpair of the symmetric matrix $H$, so that $H v_j = \lambda_j v_j$. Without loss of generality, we can assume that $v_j$ is scaled so that $\|v_j\| = 1$, i.e., $v_j^* v_j = \|v_j\|^2 = 1$. Thus

$$\lambda_j = \lambda_j (v_j^* v_j) = v_j^* (\lambda_j v_j) = v_j^* (H v_j).$$

Since $H$ is symmetric, $H = H^*$, and so

$$v_j^* (H v_j) = v_j^* H^* v_j = (H v_j)^* v_j = (\lambda_j v_j)^* v_j = \lambda_j v_j^* v_j = \lambda_j v_j.$$

Thus $\lambda_j = \lambda_j$, which is only possible if $\lambda_j$ is real. ■

**Definition 8.1** A symmetric matrix $H \in \mathbb{C}^{n \times n}$ is positive definite provided $x^* H x > 0$ for all nonzero $x \in \mathbb{C}^n$; if $x^* H x \geq 0$ for all $x \in \mathbb{C}^n$, we say $H$ is positive semidefinite.

**Theorem 8.3** All eigenvalues of a symmetric positive definite matrix are positive; all eigenvalues of a symmetric positive semidefinite matrix are nonnegative.

**Proof.** Let $(\lambda_j, v_j)$ denote an eigenpair of the symmetric positive definite matrix $H \in \mathbb{C}^{n \times n}$ with $\|v_j\|^2 = v_j^* v_j = 1$. Since $H$ is symmetric, $\lambda_j$ must be real. We conclude that

$$\lambda_j = \lambda_j v_j^* v_j = v_j^* (\lambda_j v_j) = v_j^* H v_j,$$

which must be positive since $H$ is positive definite and $v_j \neq 0$.

The proof for positive semidefinite matrices is the same, except we can only conclude that $\lambda_j = v_j^* H v_j \geq 0$. ■

### 8.2 Derivation of the singular value decomposition: Full rank case

We seek to derive the singular value decomposition of a general rectangular matrix. To simplify our initial derivation, we shall assume that $A \in \mathbb{C}^{m \times n}$ with $m \geq n$, and that $\text{rank}(A)$ is as large as possible, i.e.,

$$\text{rank}(A) = n.$$

First, form $A^* A$, which is an $n \times n$ matrix. Notice that $A^* A$ is always symmetric, since

$$(A^* A)^* = A^* (A^*)^* = A^* A.$$
Furthermore, this matrix is positive definite: notice that
\[ x'Ax = (Ax)'(Ax) = \|Ax\|^2 \geq 0. \]

Since \( \text{rank}(A) = n \), notice that
\[ \dim(N(A)) = n - \text{rank}(A) = 0. \]

Since the null space of \( A \) is trivial, \( Ax \neq 0 \) whenever \( x \neq 0 \), so
\[ x'A'Ax = \|Ax\|^2 > 0 \]
for all nonzero \( x \). Hence \( A'A \) is positive definite.

We are now ready to construct our first version of the singular value decomposition. We shall construct the pieces one at a time, then assemble them into the desired decomposition.

**Step 1. Compute the eigenvalues and eigenvectors of \( A'A \).**

As a consequence of results about symmetric matrices presented above, we can find \( n \) eigenpairs \( \{ (\lambda_j, v_j) \}_{j=1}^n \) of \( H = A'A \) with unit eigenvectors \( (v_j^*v_j = \|v_j\|^2 = 1) \) that are orthogonal to one another \( (v_j^*v_k = 0 \text{ when } j \neq k) \). We are free to pick any convenient indexing for these eigenpairs; we shall label them so that the eigenvalues are decreasing in size, \( \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n > 0 \). It is helpful to emphasize that \( v_1, \ldots, v_n \in \mathbb{C}^n \).

**Step 2. Define \( \sigma_j = \|Av_j\| = \sqrt{\lambda_j}, j = 1, \ldots, n. \)**

Note that \( \sigma_j^2 = \|Av_j\|^2 = v_j^*A'Av_j = \lambda_j \). Since the eigenvalues \( \lambda_1, \ldots, \lambda_n \) are decreasing in size, so too are the \( \sigma_j \) values:
\[ \sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n > 0. \]

**Step 3. Define \( u_j = Av_j / \sigma_j \) for \( j = 1, \ldots, n. \)**

Notice that \( u_1, \ldots, u_n \in \mathbb{C}^m \). Because \( \sigma_j = \|Av_j\| \), we ensure that
\[ \|u_j\| = \left\| \frac{1}{\sigma_j}Av_j \right\| = \frac{\|Av_j\|}{\sigma_j} = 1. \]

Furthermore, these \( u_j \) vectors are orthogonal. To see this, write
\[ u_j^*u_k = \frac{1}{\sigma_j\sigma_k} (Av_j)^*(Av_k) = \frac{1}{\sigma_j\sigma_k} v_j^*A'Av_k. \]

Since \( v_k \) is an eigenvector of \( A'A \) corresponding to eigenvalue \( \lambda_k \),
\[ \frac{1}{\sigma_j\sigma_k} v_j^*A'Av_k = \frac{1}{\sigma_j\sigma_k} v_j^*(\lambda_k v_k) = \frac{\lambda_k}{\sigma_j\sigma_k} v_j^*v_k. \]
Since the eigenvectors of the symmetric matrix $A^*A$ are orthogonal, $v_j^T v_k = 0$ when $j \neq k$, so the $u_j$ vectors inherit the orthogonality of the $v_j$ vectors:

$$u_j^T u_k = 0, \quad j \neq k.$$

**Step 4. Put the pieces together.**

For all $j = 1, \ldots, n$,

$$A v_j = \sigma_j u_j,$$

regardless of whether $\sigma_j = 0$ or not. We can stack these $n$ vector equations as columns of a single matrix equation,

$$\begin{bmatrix} Av_1 & Av_2 & \cdots & Av_n \end{bmatrix} = \begin{bmatrix} \sigma_1 u_1 & \sigma_2 u_2 & \cdots & \sigma_n u_n \end{bmatrix}.$$

Note that both matrices in this equation can be factored into the product of simpler matrices:

$$A v_1 \cdots v_n = \sigma_1 u_1 \cdots \sigma_n u_n,$$

where $A \in \mathbb{C}^{m \times n}$, $V \in \mathbb{C}^{n \times n}$, $\hat{U} \in \mathbb{C}^{m \times n}$, and $\hat{\Sigma} \in \mathbb{C}^{n \times n}$.

We now have all the ingredients for various forms of the singular value decomposition. Since the eigenvectors $v_j$ of the symmetric matrix $A^*A$ are orthonormal, the square matrix $V$ has orthonormal columns. This means that

$$V^* V = I,$$

since the $(j,k)$ entry of $V^* V$ is simply $v_j^T v_k$. Since $V$ is square, the equation $V^* V = I$ implies that $V^* = V^{-1}$. Thus, in addition to $V^* V$, we also have

$$VV^* = VV^{-1} = I.$$

Thus multiplying both sides of equation (8.2) on the right by $V^*$ gives

$$A = \hat{U} \hat{\Sigma} V^*.$$

This factorization is the reduced (or skinny) singular value decomposition of $A$. It can be obtained via the MATLAB command

```
A = svd(A, 'econ');
```
What can be said of the matrix \( \hat{U} \in \mathbb{C}^{m \times n} \)? Recall that its columns, the vectors \( u_1, \ldots, u_n \), are orthonormal. However, in contrast to \( V \), we cannot conclude that \( \hat{U} \hat{U}^* = I \) when \( m > n \). Why not? Because when \( m > n \), \( \hat{U} \) has a nontrivial null space, and hence cannot be invertible.

We wish to augment the matrix \( \hat{U} \) with \( m - n \) additional column vectors, to give a full set of \( m \) orthonormal vectors in \( \mathbb{C}^m \). Here is the recipe to find these extra vectors: For \( j = n + 1, \ldots, m \), pick \( u_j \perp \text{span}\{u_1, \ldots, u_{j-1}\} \) with \( u_j^* u_j = 1 \). Then define

\[
U = \begin{bmatrix} \vline \vline \vline \\
| & | & | \\
u_1 & \cdots & u_n \\
| & | & | \\
u_{n+1} & \cdots & u_m \\
| & | & | \\
\vline \vline \vline
\end{bmatrix} \in \mathbb{C}^{m \times m}. \quad (8.4)
\]

We have constructed \( u_1, \ldots, u_m \) to be orthonormal vectors, so \( U^* U = I \).

However, since \( U \in \mathbb{C}^{m \times m} \), this orthogonality also implies \( U^{-1} = U^* \).

Now we are ready to replace the rectangular matrix \( \hat{U} \in \mathbb{C}^{m \times n} \) in the reduced SVD (8.3) with the square matrix \( U \in \mathbb{C}^{m \times m} \). To do so, we also need to replace \( \hat{\Sigma} \in \mathbb{C}^{n \times n} \) by some \( \Sigma \in \mathbb{C}^{m \times n} \) in such a way that

\( \hat{U} \hat{\Sigma} = U \Sigma \).

The simplest approach is to obtain \( \Sigma \) by appending zeros to the end of \( \hat{\Sigma} \), thus ensuring there is no contribution when the new entries of \( U \) multiply against the new entries of \( \Sigma \):

\[
\Sigma = \begin{bmatrix} \hat{\Sigma} \\ 0 \end{bmatrix} \in \mathbb{C}^{m \times n}. \quad (8.5)
\]

Finally, we arrive at the main result, the full singular value decomposition, for the case where \( \text{rank}(A) = n \).

**Theorem 8.4 (Singular value decomposition, provisional version)**

*Suppose \( A \in \mathbb{C}^{m \times n} \) has \( \text{rank}(A) = n \), with \( m \geq n \). Then we can write*

\[
A = U \Sigma V^*,
\]

*where the columns of \( U \in \mathbb{C}^{m \times m} \) and \( V \in \mathbb{C}^{n \times n} \) are orthonormal,*

\[
U^* U = I \in \mathbb{C}^{m \times m}, \quad V^* V = I \in \mathbb{C}^{n \times n},
\]

*and \( \Sigma \in \mathbb{C}^{m \times n} \) is zero everywhere except for entries on the main diagonal, where the \((j, j)\) entry is \( \sigma_j \), for \( j = 1, \ldots, n \) and

\[
\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n > 0.
\]
The full SVD is obtained via the MATLAB command

\[ [U, S, V] = \text{svd}(A) . \]

**Definition 8.2.** Let \( A = U \Sigma V^* \) be a full singular value decomposition. The diagonal entries of \( \Sigma \), denoted \( \sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n \), are called the singular values of \( A \). The columns \( u_1, \ldots, u_m \) of \( U \) are the left singular vectors; the columns \( v_1, \ldots, v_m \) of \( V \) are the right singular vectors.

### 8.3 The dyadic form of the SVD

We are now prepared to develop an analogue of the formula (8.1) for rectangular matrices. Consider the reduced SVD,

\[ A = \hat{U} \hat{\Sigma} V^* , \]

and multiply \( \hat{U} \hat{\Sigma} \) to obtain

\[
\begin{bmatrix}
| & | & | \\
\sigma_1 & \sigma_2 & \cdots \\
| & | & | \\
| & | & | \\
\end{bmatrix}
\begin{bmatrix}
| & | & | \\
u_1 & u_2 & \cdots & u_n \\
| & | & | \\
| & | & | \\
\end{bmatrix}
= 
\begin{bmatrix}
| & | & | \\
\sigma_1 u_1 & \sigma_1 u_2 & \cdots & \sigma_n u_n \\
| & | & | \\
| & | & | \\
\end{bmatrix}.
\]

Now notice that you can write \( A = (\hat{U} \hat{\Sigma}) V^* \) as

\[
\begin{bmatrix}
| & | & | \\
\sigma_1 u_1 & \sigma_1 u_2 & \cdots & \sigma_n u_n \\
| & | & | \\
| & | & | \\
\end{bmatrix}
\begin{bmatrix}
| & | & | \\
v_1^* & v_2^* & \cdots & v_n^* \\
| & | & | \\
| & | & | \\
\end{bmatrix}
= \sum_{j=1}^{n} \sigma_j u_j v_j^* ,
\]

which parallels the form (8.1) we had for symmetric matrices:

\[ A = \sum_{j=1}^{n} \sigma_j u_j v_j^* . \quad (8.6) \]

This expression is called the *dyadic form of the SVD*. Because we have ordered \( \sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n \), the leading terms in this sum dominate the others. This fact plays a crucial role in applications where we want to approximate a matrix with its leading low-rank part.

### 8.4 A small example

Consider the matrix

\[ A = \begin{bmatrix} 1 & 1 \\ 0 & \sqrt{2} \\ \sqrt{2} & -\sqrt{2} \end{bmatrix} . \]
for which \( A^*A \) is the symmetric matrix used as an example earlier in these notes:

\[
A^*A = \begin{bmatrix}
3 & -1 \\
-1 & 3
\end{bmatrix}.
\]

This matrix has \( \text{rank}(A) = 2 = n \), so we can apply the analysis described above.

**Step 1. Compute the eigenvalues and eigenvectors of \( A^*A \).**

We have already seen that, for this matrix, \( \lambda_1 = 4 \) and \( \lambda_2 = 2 \), with

\[
v_1 = \begin{bmatrix} \sqrt{2}/2 \\ -\sqrt{2}/2 \end{bmatrix}, \quad v_2 = \begin{bmatrix} \sqrt{2}/2 \\ \sqrt{2}/2 \end{bmatrix},
\]

with \( \lambda_1 \geq \lambda_2 \), the required order. The vectors \( v_1 \) and \( v_2 \) will be the right singular vectors of \( A \).

**Step 2. Define \( \sigma_j = \|Av_j\| = \sqrt{\lambda_j} \), \( j = 1, \ldots, n \).**

In this case, we compute

\[
\sigma_1 = \sqrt{\lambda_1} = 2, \quad \sigma_2 = \sqrt{\lambda_2} = \sqrt{2}.
\]

Alternatively, we could have computed the singular values from

\[
\begin{align*}
Av_1 &= \begin{bmatrix} 1 & 1 \\
0 & 0 \\
\sqrt{2} & -\sqrt{2} \end{bmatrix} \begin{bmatrix} \sqrt{2}/2 \\
-\sqrt{2}/2 \end{bmatrix} = \begin{bmatrix} 0 \\
0 \\
2 \end{bmatrix}, \\
Av_2 &= \begin{bmatrix} 1 & 1 \\
0 & 0 \\
\sqrt{2} & -\sqrt{2} \end{bmatrix} \begin{bmatrix} \sqrt{2}/2 \\
\sqrt{2}/2 \end{bmatrix} = \begin{bmatrix} \sqrt{2} \\
0 \\
0 \end{bmatrix},
\end{align*}
\]

with \( \sigma_1 = \|Av_1\| = 2 \) and \( \sigma_2 = \|Av_2\| = \sqrt{2} \).

**Step 3. Define \( u_j = Av_j/\sigma_j \), \( j = 1, \ldots, n \).**

We use the vectors \( Av_1 \) and \( Av_2 \) computed at the last step:

\[
\begin{align*}
u_1 &= \frac{1}{\sigma_1}Av_1 = \frac{1}{2} \begin{bmatrix} 0 \\
0 \\
1 \end{bmatrix} = \begin{bmatrix} 0 \\
0 \\
1 \end{bmatrix}, \\
u_2 &= \frac{1}{\sigma_2}Av_2 = \frac{1}{\sqrt{2}} \begin{bmatrix} \sqrt{2} \\
0 \\
1 \end{bmatrix} = \begin{bmatrix} 1 \\
0 \\
0 \end{bmatrix}.
\end{align*}
\]

**Step 4. Put the pieces together.**

We immediately have the reduced SVD \( A = \hat{U}\hat{\Sigma}V^* \):

\[
\begin{bmatrix} 1 & 1 \\
0 & 0 \\
\sqrt{2} & -\sqrt{2} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\
0 & 0 \\
1 & 0 \end{bmatrix} \begin{bmatrix} 2 & 0 \\
0 & \sqrt{2} \\
\sqrt{2}/2 & \sqrt{2}/2 \end{bmatrix} \begin{bmatrix} \sqrt{2}/2 & -\sqrt{2}/2 \\
\sqrt{2}/2 & \sqrt{2}/2 \end{bmatrix}.
\]

To get the full SVD, we need a unit vector \( u_3 \) that is orthogonal to \( u_1 \) and \( u_2 \). In this case, such a vector is easy to spot:

\[
u_3 = \begin{bmatrix} 0 \\
1 \\
0 \end{bmatrix}.
\]
Thus we can write the full SVD $A = U\Sigma V^*$:

$$
\begin{bmatrix}
1 & 1 \\
0 & 0 \\
\sqrt{2} & -\sqrt{2}
\end{bmatrix}
= 
\begin{bmatrix}
0 & 1 & 0 \\
0 & 0 & 1 \\
1 & 0 & 0
\end{bmatrix}
\begin{bmatrix}
2 & 0 \\
0 & \sqrt{2} \\
\sqrt{2}/2 & -\sqrt{2}/2
\end{bmatrix}.
$$

Finally, we write the dyadic form of the SVD, $A = \sum_{j=1}^{2} \sigma_j u_j v_j^*$:

$$
\begin{bmatrix}
1 & 1 \\
0 & 0 \\
\sqrt{2} & -\sqrt{2}
\end{bmatrix}
= 
2
\begin{bmatrix}
0 & 0 & 1 \\
0 & 0 & 1 \\
\sqrt{2}/2 & -\sqrt{2}/2
\end{bmatrix}
+ 
\sqrt{2}
\begin{bmatrix}
0 & 0 \\
0 & 0 \\
\sqrt{2}/2 & \sqrt{2}/2
\end{bmatrix}.
$$

### 8.5 Derivation of the singular value decomposition: Rank deficient case

Having computed the singular value decomposition of a matrix $A \in \mathbb{C}^{m \times n}$ with rank$(A) = n$, we must now consider the adjustments necessary when rank$(A) = r < n$, still with $m \geq n$.

Recall that the dimension of the null space of $A$ is given by

$$
dim(N(A)) = n - \text{rank}(A) = n - r.
$$

How do the null spaces of $A$ and $A^*A$ compare?

**Lemma 8.1** For any matrix $A \in \mathbb{C}^{m \times n}$, $N(A^*A) = N(A)$.

**Proof.** First we show that $N(A)$ is contained in $N(A^*A)$. If $x \in N(A)$, then $Ax = 0$. Premultiplying by $A^*$ gives $A^*Ax = 0$, so $x \in N(A^*A)$.

Now we show that $N(A^*A)$ is contained in $N(A)$. If $x \in N(A^*A)$, then $A^*Ax = 0$. Premultiplying by $x^*$ gives

$$
0 = x^*A^*Ax = (Ax)^*(Ax) = \|Ax\|^2.
$$

Since $\|Ax\| = 0$, we conclude that $Ax = 0$, and so $x \in N(A)$.

Since the spaces $N(A)$ and $N(A^*A)$ each contain the other, we conclude that $N(A) = N(A^*A)$. $
$

Now we can make a crucial insight: the dimension of $N(A)$ tells us how many zero eigenvalues $A^*A$ has. In particular, suppose $x_1, \ldots, x_{n-r}$ is a basis for $N(A)$. Then $Ax_j = 0$ implies

$$
A^*Ax_j = 0, \quad j = 1, \ldots, n-r
$$

and so $\lambda = 0$ is an eigenvalue of $A^*A$ of multiplicity $n - r$.

Can you construct a $2 \times 2$ matrix $A$ whose only eigenvalue is zero, but $\dim(N(A)) = 1$? What is the multiplicity of the zero eigenvalue of $A^*A$?
How do these zero eigenvalues of \( A^*A \) affect the singular value decomposition? To begin, perform Steps 1 and 2 of the SVD procedure just as before.

**Step 1. Compute the eigenvalues and eigenvectors of \( A^*A \).**

Since we order the eigenvalues of \( A^*A \) so that \( \lambda_1 \geq \cdots \geq \lambda_n \geq 0 \), and we have just seen that zero is an eigenvalue of \( A^*A \) of multiplicity \( n - r \), we must have

\[
\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_r > 0, \quad \lambda_{r+1} = \cdots = \lambda_n = 0.
\]

The corresponding orthonormal eigenvectors are \( v_1, \ldots, v_n \), with the last \( n - r \) of these vectors in \( \mathcal{N}(A^*A) = \mathcal{N}(A) \), i.e., \( Av_j = 0 \).

**Step 2. Define \( \sigma_j = \|Av_j\| = \sqrt{\lambda_j}, j = 1, \ldots, n \).**

This step proceeds without any alterations, though now we have

\[
\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0, \quad \sigma_{r+1} = \cdots = \sigma_n = 0.
\]

The third step of the SVD construction needs alteration, since we can only define the left singular vectors via \( u = Av_j/\sigma_j \) when \( \sigma_j > 0 \), that is, for \( j = 1, \ldots, r \). Any choice for the remaining vectors, \( u_{r+1}, \ldots, u_n \), will trivially satisfy the equation \( Av_j = \sigma_j u_j \), since \( Av_j = 0 \) and \( \sigma_j = 0 \) for \( j = r + 1, \ldots, n \). Since we are building \( \hat{U} \in \mathbb{C}^{m \times n} \) (and eventually \( U \in \mathbb{C}^{m \times m} \)) to have orthonormal, we will simply build out \( u_{r+1}, \ldots, u_n \) so that all the vectors \( u_1, \ldots, u_n \) are orthonormal.

**Step 3a. Define \( u_j = Av_j/\sigma_j \) for \( j = 1, \ldots, r \).**

**Step 3b. Construct orthonormal vectors \( u_{r+1}, \ldots, u_n \).**

For each \( j = r + 1, \ldots, n \), construct a unit vector \( u_j \) such that

\[
\|u_j\| = 1, \quad u_j \perp \text{span}\{u_1, \ldots, u_{j-1}\}.
\]

This procedure is exactly the same as used above to construct the vectors \( u_{n+1}, \ldots, u_m \) to extend the reduced SVD with \( \hat{U} \in \mathbb{C}^{m \times n} \) to the full SVD with \( U \in \mathbb{C}^{m \times m} \).

**Step 4. Put the pieces together.**

This step proceeds exactly as before. Now we define

\[
\hat{U} = \begin{bmatrix}
\vdots & \cdots & \cdots & \cdots & \vdots \\
| & \cdots & \cdots & \cdots & | \\
u_1 & \cdots & u_r & u_{r+1} & \cdots & u_n \\
| & \cdots & \cdots & \cdots & | \\
\end{bmatrix} \in \mathbb{C}^{m \times n},
\]

If \( r = 0 \) (which implies the trivial case \( A = 0 \)), just set \( u_1 \) to be any unit vector.
\[ \hat{\Sigma} = \begin{bmatrix} \sigma_1 \\ & \ddots \\ & & \sigma_r \\ & & & \sigma_{r+1} \\ & & & & \ddots \\ & & & & & \sigma_n \end{bmatrix} = \begin{bmatrix} \sigma_1 \\ & \ddots \\ & & \sigma_r \\ & & & 0 \\ & & & & \ddots \\ & & & & & 0 \end{bmatrix} \in \mathbb{C}^{n \times n}, \]

and
\[ V = \begin{bmatrix} v_1 & \cdots & v_r & v_{r+1} & \cdots & v_n \end{bmatrix} \in \mathbb{C}^{n \times n}. \]

Notice that \( V \) is still a square matrix with orthonormal columns, so \( V^* V = I \) and \( V^{-1} = V^* \). Since \( Av_j = \sigma_j u_j \) holds for \( j = 1, \ldots, n \), we again have the reduced singular value decomposition
\[ A = \hat{U} \hat{\Sigma} V^*. \]

As before, \( \hat{U} \in \mathbb{C}^{m \times n} \) can be enlarged to give \( U \in \mathbb{C}^{n \times n} \) by supplying extra orthogonal unit vectors that complete a basis for \( \mathbb{C}^m \):
\[ u_j \perp \text{span}\{u_1, \ldots, u_{j-1}\}, \quad \|u_j\| = 1, \quad j = n + 1, \ldots, m. \]

Constructing \( U \in \mathbb{C}^{m \times m} \) as in (8.4) and \( \Sigma \in \mathbb{C}^{m \times n} \) as in (8.5), we have the full singular value decomposition
\[ A = U \Sigma V^*. \]

The dyadic decomposition could still be written as
\[ A = \sum_{j=1}^{n} \sigma_j u_j v_j^*, \]

but we get more insight if we crop the trivial terms from this sum. Since \( \sigma_{r+1} = \cdots = \sigma_n = 0 \), we can truncate the decomposition to its first \( r \) terms in the sum:
\[ A = \sum_{j=1}^{r} \sigma_j u_j v_j^*. \]

We will see that this form of \( A \) is especially useful for understanding the four fundamental subspaces.

### 8.6 The connection to \( AA^* \)

Our derivation of the SVD relied heavily on an eigenvalue decomposition of \( A^* A \). How does the SVD relate to \( AA^* \)? Consider forming
\[ AA^* = (U \Sigma V^*)(U \Sigma V^*)^* \]
\[ = U \Sigma V^* V \Sigma^* U^* \]
\[ = U \Sigma \Sigma^* U^*. \]
Notice that $\Sigma \Sigma^*$ is a diagonal $m \times m$ matrix:

$$\Sigma \Sigma^* = \begin{bmatrix} \hat{\Sigma} & 0 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} \hat{\Sigma}^2 & 0 \\ 0 & 0 \end{bmatrix},$$

where we have used the fact that $\hat{\Sigma}$ is a diagonal matrix. Indeed,

$$\hat{\Sigma}^2 = \begin{bmatrix} \sigma_1^2 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_n^2 \end{bmatrix} = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \cdots \lambda_n \end{bmatrix},$$

where the $\lambda_j$ values still denote the eigenvalues of $A^*A$. Thus equation (8.7) becomes

$$AA^* = U \begin{bmatrix} \Lambda & 0 \\ 0 & 0 \end{bmatrix} U^*,$$

which is a diagonalization of $AA^*$. Postmultiplying this equation by $U$, we have

$$(AA^*)U = U \begin{bmatrix} \Lambda & 0 \\ 0 & 0 \end{bmatrix};$$

the first $n$ columns of this equation give

$$AA^* u_j = \lambda_j u_j, \quad j = 1, \ldots, n,$$

while the last $m - n$ columns give

$$AA^* u_j = 0 u_j, \quad j = n + 1, \ldots, m.$$

Thus the columns $u_1, \ldots, u_n$ are eigenvectors of $AA^*$. Notice then that $AA^*$ and $A^*A$ have the same eigenvalues, except that $AA^*$ has $m - n$ extra zero eigenvalues.

### 8.7 Modification for the case of $m < n$

How does the singular value decomposition change if $A$ has more columns than rows, $n > m$? The answer is easy: write the SVD of $A^*$ (which has more rows than columns) using the procedure above, then take the conjugate-transpose of each term in the SVD. If this makes good sense, skip ahead to the next section. If you prefer the gory details, read on.

We will formally adapt the steps described above to handle the case $n > m$. Let $r = \text{rank}(A) \leq m$.

**Step 1. Compute the eigenvalues and eigenvectors of $AA^*$**.

Label the eigenvalues of $AA^* \in \mathbb{C}^{m \times m}$ as

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_m$$

and corresponding orthonormal eigenvectors as

$$u_1, u_2, \ldots, u_m.$$
Step 2. Define \( \sigma_j = \| A^* u_j \| = \sqrt{\lambda_j}, j = 1, \ldots, m. \)

Step 3a. Define \( v_j = A^* u_j / \sigma_j \) for \( j = 1, \ldots, r. \)

Step 3b. Construct orthonormal vectors \( v_{r+1}, \ldots, v_m. \)

Notice that these vectors only arise in the rank-deficient case, when \( r < m. \)

Step 3c. Construct orthonormal vectors \( v_{m+1}, \ldots, v_n. \)

Following the same procedure as step 3b, we construct the extra vectors needed to obtain a full orthonormal basis for \( \mathbb{C}^n. \)

Step 4. Put the pieces together.

First, defining

\[
\begin{align*}
\hat{U} &= \begin{bmatrix} u_1 & \cdots & u_m \end{bmatrix} \in \mathbb{C}^{m \times m}, \\
\hat{V} &= \begin{bmatrix} v_1 & \cdots & v_m \end{bmatrix} \in \mathbb{C}^{n \times m},
\end{align*}
\]

with diagonal matrix

\[
\hat{\Sigma} = \text{diag}(\sigma_1, \ldots, \sigma_m) \in \mathbb{C}^{m \times m},
\]

we have the reduced SVD

\[
A = U \hat{\Sigma} \hat{V}^*.
\]

To obtain the full SVD, we extend \( \hat{V} \) to obtain

\[
V = \begin{bmatrix} v_1 & \cdots & v_m & v_{m+1} & \cdots & v_n \end{bmatrix} \in \mathbb{C}^{n \times n},
\]

and similarly extend \( \hat{\Sigma}, \)

\[
\Sigma = \begin{bmatrix} \hat{\Sigma} & 0 \end{bmatrix} \in \mathbb{C}^{m \times n},
\]

where we have now added extra zero columns, in contrast to the extra zero rows added in the \( m > n \) case in (8.5). We thus arrive at the full SVD,

\[
A = U \Sigma V^*.
\]

8.8 General statement of the singular value decomposition

We now can state the singular value decomposition in its fullest generality.
Theorem 8.5 (Singular value decomposition) Suppose $A \in \mathbb{C}^{m \times n}$ has $\text{rank}(A) = r$. Then we can write

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

where the columns of $U \in \mathbb{C}^{m \times m}$ and $V \in \mathbb{C}^{n \times n}$ are orthonormal,

$$U^*U = I \in \mathbb{C}^{m \times m}, \quad V^*V = I \in \mathbb{C}^{n \times n},$$

and $\Sigma \in \mathbb{C}^{m \times n}$ is zero everywhere except for entries on the main diagonal, where the $(j,j)$ entry is $\sigma_j$, for $j = 1, \ldots, \min\{m, n\}$ and

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > \sigma_{r+1} = \cdots = \sigma_{\min\{m,n\}} = 0.$$ 

Denoting the columns of $U$ and $V$ as $u_1, \ldots, u_m$ and $v_1, \ldots, v_m$, we can write

$$A = \sum_{j=1}^{r} \sigma_j u_j v_j^*, \quad (8.8)$$

Of course, when $r = 0$ all the singular values are zero; when $r = \min\{m, n\}$, all the singular values are positive.

8.9 Connection to the four fundamental subspaces

Having labored to develop the singular value decomposition in its complete generality, we are ready to reap its many rewards. We begin by establishing the connection between the singular vectors and the ‘four fundamental subspaces,’ i.e., the column space

$$\mathcal{R}(A) = \{ Ax : x \in \mathbb{C}^n \} \subseteq \mathbb{C}^m,$$

the row space

$$\mathcal{R}(A^*) = \{ A^*y : y \in \mathbb{C}^m \} \subseteq \mathbb{C}^n,$$

the null space

$$\mathcal{N}(A) = \{ x \in \mathbb{C}^n : Ax = 0 \} \subseteq \mathbb{C}^n,$$

and the left null space

$$\mathcal{N}(A^*) = \{ y \in \mathbb{C}^m : A^*y = 0 \} \subseteq \mathbb{C}^m.$$ 

We shall explore these spaces using the dyadic form of the SVD (8.8). To characterize the column space, apply $A$ to a generic vector $x \in \mathbb{C}^n$:

$$Ax = \left( \sum_{j=1}^{r} \sigma_j u_j v_j^* \right) x = \sum_{j=1}^{r} (\sigma_j u_j v_j^* x) = \sum_{j=1}^{r} (\sigma_j v_j^* x) u_j, \quad (8.9)$$

where in the last step we have switched the order of the scalar $v_j^* x$ and the vector $u_j$. We see that $Ax$ is a weighted sum of the vectors $u_1, \ldots, u_r$. Since this must hold for all $x \in \mathbb{C}^n$, we conclude that

$$\mathcal{R}(A) \subseteq \text{span}\{ u_1, \ldots, u_r \}. $$
Can we conclude the converse? We know that \( \mathcal{R}(A) \) is a subspace, so if we can show that each of the vectors \( u_1, \ldots, u_r \) is in \( \mathcal{R}(A) \), then we will know that

\[
\text{span}\{u_1, \ldots, u_r\} \subseteq \mathcal{R}(A).
\]  

(8.10)

To show that \( u_k \in \mathcal{R}(A) \), we must find some \( x \) such that \( Ax = u_k \). Inspect equation (8.9). We can make \( Ax = u_k \) if all the coefficients \( \sigma_j v_j^* x \) are zero when \( j \neq k \), and \( \sigma_k v_k^* x = 1 \). Can you see how to use orthogonality of the right singular vectors \( v_1, \ldots, v_r \) to achieve this? Setting

\[
x = \frac{1}{\sigma_k} v_k^*,
\]

we have \( Ax = u_k \). Thus \( u_k \in \mathcal{R}(A) \), and we can conclude that (8.10) holds. Since \( \mathcal{R}(A) \) and \( \text{span}\{u_1, \ldots, u_r\} \) contain one another, we conclude that

\[
\mathcal{R}(A) = \text{span}\{u_1, \ldots, u_r\}.
\]

We can characterize the row space in exactly the same way, using the dyadic form

\[
A^* = \left( \sum_{j=1}^r \sigma_j u_j v_j^* \right)^* = \sum_{j=1}^r \left( \sigma_j u_j v_j^* \right)^* = \sum_{j=1}^r \sigma_j v_j^* u_j^*.
\]

Adapting the argument we have just made leads to

\[
\mathcal{R}(A^*) = \text{span}\{v_1, \ldots, v_r\}.
\]

Equation (8.9) for \( Ax \) is also the key that unlocks the null space \( \mathcal{N}(A) \). For what \( x \in \mathbb{C}^n \) does \( Ax = 0 \)? Let us consider

\[
\|Ax\|^2 = (Ax)^*(Ax) = \left( \sum_{j=1}^r (\sigma_j v_j^*) x u_j \right) \left( \sum_{k=1}^r \sigma_k v_k^* x u_k \right)
\]

\[
= \left( \sum_{j=1}^r (\sigma_j x^* v_j) u_j^* \right) \left( \sum_{k=1}^r \sigma_k v_k^* x u_k \right)
\]

\[
= \sum_{j=1}^r \sum_{k=1}^r \left( \sigma_j x^* v_j \right) \left( \sigma_k v_k^* x \right) u_j^* u_k.
\]

Since the left singular vectors are orthogonal, \( u_j^* u_k = 0 \) for \( j \neq k \), this double-sum collapses: only the terms with \( j = k \) make a nontrivial contribution:

\[
\|Ax\|^2 = \sum_{j=1}^r (\sigma_j x^* v_j) (\sigma_j v_j^* x) u_j^* u_j = \sum_{j=1}^r \sigma_j^2 |v_j^* x|^2, \tag{8.11}
\]

since \( u_j^* u_j = 1 \) and \( (x^* v_j)(v_j^* x) = |v_j^* x|^2 \). If \( z \) is complex, then \( z^* z = zz = |z|^2 \). We quietly used \( (v_j^* x)^* = x^* v_j \) here. If \( v_j^* x \) is real, \( (v_j^* x)^* = x^* v_j = v_j^* x \); if \( v_j^* x \) is complex, we must be more careful: \( (v_j^* x)^* = x^* v_j = \overline{v_j^* x} \), where the line denotes complex-conjugation.
Since \( \sigma_j > 0 \), the right-hand side of (8.11) is the sum of nonnegative numbers. To have \( \|Ax\| = 0 \), all the coefficients in this sum must be zero. The only way for that to happen is for \( v_j^T x = 0, \quad j = 1, \ldots, r \), i.e., \( Ax = 0 \) if and only if \( x \) is orthogonal to \( v_1, \ldots, v_r \). We already have a characterization of such vectors from the singular value decomposition:

\[
x \in \text{span}\{v_{r+1}, \ldots, v_n\}.
\]

Thus we conclude

\[
N(A) = \text{span}\{v_{r+1}, \ldots, v_n\}.
\]

To compute \( N(A^*) \), we can repeat the same argument based on \( \|A^* y\|^2 \) to obtain

\[
N(A^*) = \text{span}\{u_{r+1}, \ldots, u_m\}.
\]

Putting these results together, we arrive at a beautiful elaboration of the Fundamental Theorem of Linear Algebra\(^1\).

**Theorem 8.6 (Fundamental Theorem of Linear Algebra, SVD Version)**

Suppose \( A \in \mathbb{C}^{m \times n} \) has \( \text{rank}(A) = r \), with left singular vectors \( \{u_1, \ldots, u_m\} \) and right singular vectors \( \{v_1, \ldots, v_n\} \). Then

\[
\mathcal{R}(A) = \text{span}\{u_1, \ldots, u_r\},
\]

\[
N(A^*) = \text{span}\{u_{r+1}, \ldots, u_m\},
\]

\[
\mathcal{R}(A^*) = \text{span}\{v_1, \ldots, v_r\},
\]

\[
N(A) = \text{span}\{v_{r+1}, \ldots, v_n\},
\]

which implies

\[
\mathcal{R}(A) \oplus N(A^*) = \text{span}\{u_1, \ldots, u_m\} = \mathbb{C}^m
\]

\[
\mathcal{R}(A^*) \oplus N(A) = \text{span}\{v_1, \ldots, v_n\} = \mathbb{C}^n,
\]

and

\[
\mathcal{R}(A) \perp N(A^*), \quad \mathcal{R}(A^*) \perp N(A).
\]

### 8.10 Matrix norms

How ‘large’ is a matrix? We do not mean dimension – but how large, in aggregate, are its entries? One can imagine a multitude of ways to measure the entries; perhaps most natural is to sum the squares of

\[\text{Matrix norms}\]
the entries, then take the square root. This idea is useful, but we prefer a more subtle alternative that is of more universal utility throughout mathematics: we shall gauge the size \( \mathbf{A} \in \mathbb{C}^{m \times n} \) by the maximum amount it can stretch a vector, \( \mathbf{x} \in \mathbb{C}^{n} \). That is, we will measure \( \| \mathbf{A} \| \) by the largest that \( \| \mathbf{A}\mathbf{x} \| \) can be. Of course, we can inflate \( \| \mathbf{A}\mathbf{x} \| \) as much as we like simply by making \( \| \mathbf{x} \| \) larger, which we avoid by imposing a normalization: \( \| \mathbf{x} \| = 1 \). We arrive at the definition

\[
\| \mathbf{A} \| = \max_{\| \mathbf{x} \| = 1} \| \mathbf{A}\mathbf{x} \|.
\]

To study \( \| \mathbf{A}\mathbf{x} \| \), we could appeal to the formula (8.11); however, we will take a slightly different approach. First, suppose that \( \mathbf{Q} \) is some matrix with orthonormal columns, so that \( \mathbf{Q}^\top\mathbf{Q} = \mathbf{I} \). Then

\[
\| \mathbf{Q}\mathbf{x} \|^2 = (\mathbf{Q}\mathbf{x})^\top(\mathbf{Q}\mathbf{x}) = \mathbf{x}^\top\mathbf{Q}^\top\mathbf{Q}\mathbf{x} = \mathbf{x}^\top\mathbf{x} = \| \mathbf{x} \|^2,
\]

so premultiplying by \( \mathbf{Q} \) does not alter the norm of \( \mathbf{x} \). Now substitute the full \( \mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top \) for \( \mathbf{A} \):

\[
\| \mathbf{A}\mathbf{x} \| = \| \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top\mathbf{x} \| = \| \mathbf{\Sigma}\mathbf{V}^\top\mathbf{x} \|,
\]

where we have used the orthonormality of the columns of \( \mathbf{U} \). Now define a new variable \( \mathbf{y} = \mathbf{V}^\top\mathbf{x} \) (which means \( \mathbf{V}\mathbf{y} = \mathbf{x} \)), and notice that

\[
\| \mathbf{x} \| = \| \mathbf{V}^\top\mathbf{x} \| = \| \mathbf{y} \|,
\]

since \( \mathbf{V} \) is a square matrix with orthonormal columns (and hence orthonormal rows). Now we can compute the matrix norm:

\[
\| \mathbf{A} \| = \max_{\| \mathbf{x} \| = 1} \| \mathbf{A}\mathbf{x} \| = \max_{\| \mathbf{y} \| = 1} \| \mathbf{\Sigma}\mathbf{y} \| = \max_{\| \mathbf{y} \| = 1} \| \mathbf{\Sigma}\mathbf{y} \|
\]

So the norm of \( \mathbf{A} \) is the same as the norm of \( \mathbf{\Sigma} \). We now must figure out how to pick the unit vector \( \mathbf{y} \) to maximize \( \| \mathbf{\Sigma}\mathbf{y} \| \). This is easy: we want to optimize

\[
\| \mathbf{\Sigma}\mathbf{y} \|^2 = \sigma_1^2|y_1|^2 + \cdots + \sigma_r^2|y_r|^2
\]

subject to \( 1 = \| \mathbf{y} \|^2 \geq |y_1|^2 + \cdots + |y_r|^2 \). Since \( \sigma_1 \geq \cdots \geq \sigma_r \),

\[
\| \mathbf{\Sigma}\mathbf{y} \|^2 = \sigma_1^2|y_1|^2 + \cdots + \sigma_r^2|y_r|^2
\]

\[
\leq \sigma_1^2(|y_1|^2 + \cdots + |y_r|^2) \leq \sigma_1^2\| \mathbf{y} \|^2 = \sigma_1^2,
\]

resulting in the upper bound

\[
\| \mathbf{\Sigma} \| = \max_{\| \mathbf{y} \| = 1} \| \mathbf{\Sigma}\mathbf{y} \| \leq \sigma_1. \tag{8.12}
\]

Will any unit vector \( \mathbf{y} \) attain this upper bound? That is, can we find such a vector so that \( \| \mathbf{\Sigma}\mathbf{y} \| = \sigma_1 \)? Sure: just take \( \mathbf{y} = [1, 0, \cdots, 0]^\top \) to be the first column of the identity matrix. For this special vector,

\[
\| \mathbf{\Sigma}\mathbf{y} \|^2 = \sigma_1^2|y_1|^2 + \cdots + \sigma_r^2|y_r|^2 = \sigma_1^2.
\]
Since $\|\Sigma y\|$ can be no larger than $\sigma_1$ for any $y$, and since $\|\Sigma y\| = \sigma_1$ for at least one choice of $y$, we conclude

$$\|\Sigma\| = \max_{\|y\|=1} \|\Sigma y\| = \sigma_1,$$

and hence the norm of a matrix is its largest singular value:

$$\|A\| = \sigma_1.$$

Consider the matrix

$$A = \begin{bmatrix} 1/2 & 1 \\ -1/2 & 1 \end{bmatrix} = \left( \frac{\sqrt{2}}{2} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \right) \begin{bmatrix} \sqrt{2} & 0 \\ 0 & \sqrt{2}/2 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}^*.$$

We see from this SVD that $\|A\| = \sigma_1 = \sqrt{2}$. For this example the vector $Ax$ has the form

$$Ax = \sigma_1 (v_1^* x) u_1 + \sigma_2 (v_2^* x) u_2 = \sqrt{2} x_2 u_1 + \frac{\sqrt{2}}{2} x_1 u_2,$$

so $Ax$ is a blend of some expansion in the $u_1$ direction and some contraction in the $u_2$ direction. We maximize the size of $Ax$ by picking an $x$ for which $Ax$ is maximally rich in $u_1$, i.e., $x = v_1$.

### 8.11 Low-rank approximation

Perhaps the most important property of the singular value decomposition is its ability to immediately deliver optimal low-rank approximations to a matrix. The dyadic form

$$A = \sum_{j=1}^{r} \sigma_j u_j v_j^*$$

writes the rank-$r$ matrix $A$ as the sum of the $r$ rank-1 matrices

$$\sigma_j u_j v_j^*.$$

Since $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0$, we might hope that the partial sum

$$\sum_{j=1}^{k} \sigma_j u_j v_j^*$$

will give a good approximation to $A$ for some value of $k$ that is much smaller than $r$ (mathematicians write $k \ll r$ for emphasis). This is especially true in situations where $A$ models some low-rank phenomenon, but some noise (such as random sampling errors, when the entries of $A$ are measured from some physical process) causes $A$...
to have much larger rank. If the noise is small relative to the “true”
data in \( A \), we expect \( A \) to have a number of very small singular val-
ues that we might wish to neglect as we work with \( A \). We will see
examples of this kind of behavior in the next chapter.

For square diagonalizable matrices, the eigenvalue decompositions
we wrote down in Chapter 6 also express \( A \) as the sum of rank-1
matrices,

\[
A = \sum_{j=1}^{n} \lambda_j w_j \hat{w}_j^* ,
\]

but there are three key distinctions that make the singular value
decomposition a better tool for developing low-rank approximations
to \( A \).

1. The SVD holds for all matrices, while the eigenvalue decomposi-
tion only holds for square matrices.

2. The singular values are nonnegative real numbers whose ordering
\[
\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0
\]
gives a natural way to understand how much the rank-1 matrices
\( \sigma_j u_j v_j^* \) contribute to \( A \). In contrast, the eigenvalues will generally
be complex numbers, and thus do not have the same natural order.

3. The eigenvectors are not generally orthogonal, and this can skew
the rank-1 matrices \( \lambda_j w_j \hat{w}_j^* \) away from giving good approxima-
tions to \( A \). In particular, we can find that \( \| w_j \hat{w}_j^* \| \gg 1 \), whereas
the matrices \( u_j v_j^* \) from the SVD always satisfy \( \| u_j v_j^* \| = 1 \).

This last point is subtle, so let us investigate it with an example.
Consider

\[
A = \begin{bmatrix}
2 & 100 \\
0 & 1
\end{bmatrix}
\]

with eigenvalues \( \lambda_1 = 2 \) and \( \lambda_2 = 1 \) and eigenvalue decomposition

\[
A = \mathbf{W} \Delta \mathbf{W}^{-1} = \begin{bmatrix}
1 & 1 \\
0 & -1/100
\end{bmatrix} \begin{bmatrix}
2 & 0 \\
0 & 1
\end{bmatrix} \begin{bmatrix}
1 & 100 \\
0 & -100
\end{bmatrix}
\]

\[
= \lambda_1 w_1 \hat{w}_1^* + \lambda_2 w_2 \hat{w}_2^*
\]

\[
= 2 \begin{bmatrix}
1 \\
0
\end{bmatrix} \begin{bmatrix}
1 & 100 \\
0 & -1/100
\end{bmatrix} + 1 \begin{bmatrix}
1 \\
-1/100
\end{bmatrix} \begin{bmatrix}
0 & -100 \\
0 & 1
\end{bmatrix}
\]

\[
= 2 \begin{bmatrix}
1 \\
0
\end{bmatrix} \begin{bmatrix}
100 \\
0
\end{bmatrix} + 1 \begin{bmatrix}
0 \\
0
\end{bmatrix} \begin{bmatrix}
-100 \\
1
\end{bmatrix} .
\]

Let us inspect individually the two rank-1 matrices that appear in the
eigendecomposition:

\[
\lambda_1 w_1 \hat{w}_1^* = \begin{bmatrix}
2 & 200 \\
0 & 0
\end{bmatrix}, \quad \lambda_2 w_2 \hat{w}_2^* = \begin{bmatrix}
0 & -100 \\
0 & 1
\end{bmatrix} .
\]
Neither matrix individually gives a good approximation to \( A \):
\[
A - \lambda_1 w_1 \hat{w}_1^* = \begin{bmatrix} 0 & -100 \\ 0 & 1 \end{bmatrix}, \quad A - \lambda_2 w_2 \hat{w}_2^* = \begin{bmatrix} 2 & 200 \\ 0 & 0 \end{bmatrix}.
\]
Both rank-1 “approximations” to \( A \) leave large errors!

Contrast this situation with the rank-1 approximation \( \sigma_1 u_1 v_1^* \) given by the SVD for this \( A \). To five decimal digits, we have
\[
A = U \Sigma V^* = \begin{bmatrix} 0.99995 & -0.01000 \\ 0.01000 & 0.99995 \end{bmatrix} \begin{bmatrix} 100.025 & 0 \\ 0 & 0.020 \end{bmatrix} \begin{bmatrix} 0.01999 & 0.99980 \\ -0.99880 & 0.19999 \end{bmatrix}
\]
\[
= \sigma_1 u_1 v_1^* + \sigma_2 u_2 v_2^*
\]
\[
= 100.025 \begin{bmatrix} 0.99995 \\ 0.01000 \end{bmatrix} \begin{bmatrix} 0.01999 & 0.99980 \\ -0.99880 & 0.19999 \end{bmatrix} + 0.020 \begin{bmatrix} 0.01999 \\ 0.00020 \end{bmatrix} \begin{bmatrix} 0.00999 & -0.00020 \\ -0.99975 & 0.00040 \end{bmatrix}.
\]
Like the eigendecomposition, the SVD breaks \( A \) into two rank-1 pieces:
\[
\sigma_1 u_1 v_1^* = \begin{bmatrix} 1.99980 & 100.00000 \\ 0.01999 & 0.99960 \end{bmatrix}, \quad \sigma_2 u_2 v_2^* = \begin{bmatrix} 0.00020 & 0.00000 \\ -0.01999 & 0.00040 \end{bmatrix}.
\]
The first of these, the dominant term in the SVD, gives an excellent approximation to \( A \):
\[
A - \sigma_1 u_1 v_1^* = \begin{bmatrix} 0.00020 & 0.00000 \\ -0.01999 & 0.00040 \end{bmatrix}.
\]
The key factor making this approximation so good is that \( \sigma_1 \gg \sigma_2 \). What is more remarkable is that the dominant part of the singular value decomposition is actually the best low-rank approximation for all matrices.

**Definition 8.3** Let \( A = \sum_{j=1}^r \sigma_j u_j v_j^* \) be a rank-\( r \) matrix, written in terms of its singular value decomposition. Then for any \( k \leq r \), the truncated singular value of rank-\( k \) is the partial sum
\[
A_k = \sum_{j=1}^k \sigma_j u_j v_j^*.
\]

**Theorem 8.7 (Schmidt–Mirsky–Eckart–Young)** Let \( A \in \mathbb{C}^{m \times n} \). Then for all \( k \leq \text{rank}(A) \), the truncated singular value decomposition
\[
A_k = \sum_{j=1}^k \sigma_j u_j v_j^*
\]
is a best rank-\( k \) approximation to \( A \), in the sense that
\[
\|A - A_k\| = \min_{\text{rank}(X) \leq k} \|A - X\| = \sigma_{k+1}.
\]
It is easy to see that this $A_k$ gives the approximation error $\sigma_{k+1}$, since

$$A - A_k = \sum_{j=k+1}^{r} \sigma_j u_j v_j^T = \sum_{j=k+1}^{r} \sigma_j u_j v_j^T,$$

and this last expression is an SVD for the error in the approximation $A - A_k$. As described in Section 8.10, the norm of a matrix equals its largest singular value, so

$$\| A - A_k \| = \| \sum_{j=k+1}^{r} \sigma_j u_j v_j^T \| = \sigma_{k+1}.$$

To complete the proof, one needs to show that no other rank-$k$ matrix can come closer to $A$ than $A_k$. This pretty argument is a bit too intricate for this course, but we include it in the margin for those that are interested.

### 8.11.1 Compressing images with low rank approximations

Image compression provides the most visually appealing application of the low-rank matrix factorization ideas we have just described. An image can be represented as a matrix. For example, typical grayscale images consist of a rectangular array of pixels, $m$ in the vertical direction, $n$ in the horizontal direction. The color of each of those pixels is denoted by a single number, an integer between 0 (black) and 255 (white). (This gives $2^8 = 256$ different shades of gray for each pixel. Color images are represented by three such matrices: one for red, one for green, and one for blue. Thus each pixel in a typical color image takes $(2^8)^3 = 2^{24} = 16,777,216$ shades.)

**MATLAB** has many built-in routines for processing images. The `imread` command reads in image files. For example, if you want to load the file `snapshot.jpg` into **MATLAB**, you would use the command:

```matlab
A = double(imread('snapshot.jpg'));
```

If your file contains a grayscale image, $A$ will now contain the $m \times n$ matrix containing the gray colors of your image. If you have a color image, then $A$ will be an $m \times n \times 3$ matrix, and you will need to extract the color levels in an extra step.

```matlab
Ared = A(:,:,1); Agreen = A(:,:,2); Ablue = A(:,:,3);
```

The `double` command converts the entries of the image into floating point numbers. (To conserve memory, **MATLAB**’s default is to save the entries of an image as integers, but **MATLAB**’s linear algebra routines like `svd` will only work with floating point matrices.) Finally, to visualize an image in **MATLAB**, use

```matlab
imagesc(A)
```

Let $X \in \mathbb{C}^{m \times n}$ be any rank-$k$ matrix. The Fundamental Theorem of Linear Algebra gives $\mathbb{C}^n = \mathcal{R}(X^T) \oplus \mathcal{N}(X)$. Since $\text{rank}(X^T) = \text{rank}(X) = k$, notice that $\dim(\mathcal{N}(X)) = n - k$. From the singular value decomposition of $A$ extract $v_1, \ldots, v_{k+1}$, a basis for some $k + 1$ dimensional subspace of $\mathbb{C}^n$. Since $\mathcal{N}(X) \subseteq \mathbb{C}^n$ has dimension $n - k$, it must be that the intersection

$$\mathcal{N}(X) \cap \mathcal{N}(v_1, \ldots, v_{k+1})$$

has dimension at least one. (Otherwise, $\mathcal{N}(X) \oplus \mathcal{N}(v_1, \ldots, v_{k+1})$ would be an $n + 1$ dimensional subspace of $\mathbb{C}^n$—impossible!) Let $z$ be some unit vector in that intersection: $\| z \| = 1$ and

$$z \in \mathcal{N}(X) \cap \mathcal{N}(v_1, \ldots, v_{k+1}).$$

Expand $z = \gamma_1 v_1 + \cdots + \gamma_{k+1} v_{k+1}$, so that $\| z \| = 1$ implies

$$1 = z^* z = \left( \sum_{j=1}^{k+1} \gamma_j v_j^* \right) \left( \sum_{j=1}^{k+1} \gamma_j v_j \right) = \sum_{j=1}^{k+1} | \gamma_j |^2.$$

Since $z \in \mathcal{N}(X)$, we have

$$\| A - X \| \geq \| (A - X) z \| = \| A z \|,$$

and then

$$\| A z \| = \left\| \sum_{j=1}^{k+1} \gamma_j v_j z \right\| = \left\| \sum_{j=1}^{k+1} \gamma_j u_j \right\|.$$

Since $\sigma_{k+1} \leq \sigma_k \leq \cdots \leq \sigma_1$ and the $u_j$ vectors are orthogonal,

$$\left\| \sum_{j=1}^{k+1} \gamma_j u_j \right\|_2 \geq \sigma_{k+1} \left\| \sum_{j=1}^{k+1} \gamma_j u_j \right\|_2.$$

But notice that

$$\left\| \sum_{j=1}^{k+1} | \gamma_j |^2 \right\|_2 = \sum_{j=1}^{k+1} | \gamma_j |^2 = 1,$$

where the last equality was derived above from the fact that $\| z \|_2 = 1$. In conclusion, for any rank-$k$ matrix $X$,

$$\| A - X \|_2 \geq \sigma_{k+1} \left\| \sum_{j=1}^{k+1} \gamma_j u_j \right\|_2 = \sigma_{k+1}.$$

(This proof is adapted from §3.2.3 of Demmel’s text.)

and, if the image is grayscale, follow this with

```
colormap(gray)
```

The `imagesc` command is a useful tool for visualizing any matrix of data; it does not require that the entries in \( A \) be integers. (However, for color images stored in \( m \times n \times 3 \) floating point matrices, you need to use `imagesc(uint8(A))` to convert \( A \) back to positive integer values.)

Images are ripe for data compression: Often they contain broad regions of similar colors, and in many areas of the image adjacent rows (or columns) will look quite similar. If the image stored in \( A \) can be represented well by a rank-\( k \) matrix, then one can approximate \( A \) by storing only the leading \( k \) singular values and vectors. To build this approximation

\[
A_k = \sum_{j=1}^{k} \sigma_j u_j v_j^*,
\]

one need only store \( k(1 + m + n) \) values. When \( k(1 + m + n) \ll mn \), there will be a significant savings in storage, thus giving an effective compression of \( A \).

Let us look at an example to see how effective this image compression can be. For convenience we shall use an image built into MATLAB,

```
load gatlin, A = X;
imagesc(A), colormap(gray)
```

which shows some of the key developers of the numerical linear alge-

![original uncompressed image, rank = 480](image)

Figure 8.1: A sample image: the founders of numerical linear algebra at an early Gatlinburg Symposium. From left to right: Jim Wilkinson, Wallace Givens, George Forsythe, Alston Householder, Peter Henrici, and Friedrich Bauer.
bra algorithms we have studied this semester, gathered in Gatlinburg, Tennessee, for an important early conference in the field. The image is of size $480 \times 640$, so $\text{rank}(A) \leq 480$. We shall compress this image with truncated singular value decompositions. Figures 8.2 and 8.3 show compressions of $A$ for dimensions ranging from $k = 200$ down to $k = 1$. For $k = 200$ and 100, the compression $A_k$ provides an excellent proxy for the full image $A$. For $k = 50, 25$ and 10, the quality degrades a bit, but even for $k = 10$ you can still tell that the image shows six men in suits standing on a patterned floor. For $k \leq 5$ we lose much of the quality, but isn’t it remarkable how much structure is still apparent even when $k = 5$? The last image is interesting as

\begin{align*}
\text{truncated SVD, rank $k = 200$} & \quad \text{truncated SVD, rank $k = 100$} \\
\text{truncated SVD, rank $k = 50$} & \quad \text{truncated SVD, rank $k = 25$}
\end{align*}

Figure 8.2: Compressions of the Gatlinburg image in Figure 8.1 using truncated SVDs $A_k = \sum_{j=1}^{k} \sigma_j u_j v_j^\ast$. Each of these images can be stored with less memory than the original full image.
Figure 8.3: Continuation of Figure 8.3, showing compressions of the Gatlinburg image via truncated SVDs of rank 10, 5, 2, and 1. The rank-10 image might be useful as a “thumbnail” sketch of the image (e.g., an icon on a computer desktop), but the other images are compressed beyond the point of being useful.

a visualization of a rank-1 matrix: each row is a multiple of all the other rows, and each column is a multiple of all the other columns.

We gain an understanding of the quality of this compression by looking at the singular values of $A$, shown in Figure 8.4. The first singular value $\sigma_1$ is about an order of magnitude larger than the rest, and the singular values decay quite rapidly. (Notice the logarithmic vertical axis.) We have $\sigma_1 \approx 15,462$, while $\sigma_{50} \approx 204.48$. When we truncate the singular value decomposition at $k = 50$, the neglected terms in the singular value decomposition do not make a major contribution to the image.
8. The Singular Value Decomposition

Figure 8.4: Singular values of the $640 \times 480$ Gatlinburg image matrix. The first few singular values are much larger than the rest, suggesting the potential for accurate low-rank approximation (compression).

STUDENT EXPERIMENTS

8.1. The two images shown in Figure 8.5 show characters generated in the MinionPro italic font, defined in the image files minionamp.jpg and minion11.jpg, each of which leads to a $200 \times 200$ matrix. Which image do you think will better lend itself to low-rank approximation? Compute the singular values and truncated SVD approximations for a variety of ranks $k$. Do your results agree with your intuition?

Figure 8.5: Two images showing characters in the italic MinionPro font. (The first is an ampersand, which one can clearly see here derives from the Latin word et, meaning “and.”)

8.12 Special Topic: Reducing dimensions with POD

8.13 Afterword

The singular value decomposition was developed in its initial form by Eugenio Beltrami (1873) and, independently, by Camille Jordan (1874).\textsuperscript{2}