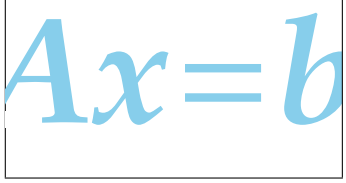
Matrix Methods for Computational Modeling and Data Analytics

Virginia Tech · Spring 2019

Mark Embree
embree@vt.edu



version of 11 April 2019

Chapter 6
The Singular Value Decomposition

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.

6.1 Eigenvalues and eigenvectors of symmetric matrices

To derive the singular value decomposition of a general (rectangular) matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$, we shall rely on several special properties of the square, symmetric matrix $\mathbf{A}^T\mathbf{A}$. While this course assumes you are well acquainted with eigenvalues and eigenvectors, we will recall some fundamental concepts, especially pertaining to symmetric matrices.

6.1.1 A passing nod to complex numbers

Recall that even if a matrix has real number entries, it could have eigenvalues that are complex numbers; the corresponding eigenvectors will also have complex entries. Consider, for example, the matrix

$$\mathbf{S} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}.$$

To find the eigenvalues of $\mathbf{S}$, form the *characteristic polynomial*

$$\det(\lambda\mathbf{I} - \mathbf{S}) = \det\left(\begin{bmatrix} \lambda & 1 \\ -1 & \lambda \end{bmatrix}\right) = \lambda^2 + 1.$$

Factor this polynomial (e.g., using the quadratic formula) to get

$$\det(\lambda\mathbf{I} - \mathbf{S}) = \lambda^2 + 1 = (\lambda - \mathrm{i})(\lambda + \mathrm{i}),$$

where $\mathrm{i} = \sqrt{-1}$. Thus, we conclude that $\mathbf{S}$ (a matrix with *real* entries) has the *complex* eigenvalues

$$\lambda_1 = \mathrm{i}, \qquad \lambda_2 = -\mathrm{i}$$

with corresponding eigenvectors

$$\mathbf{v}_1 = \begin{bmatrix} \mathrm{i} \\ 1 \end{bmatrix}, \qquad \mathbf{v}_2 = \begin{bmatrix} -\mathrm{i} \\ 1 \end{bmatrix}.$$

To find the eigenvector associated with λ_1, we need to find some nonzero $\mathbf{v} \in \mathcal{N}(\lambda_1\mathbf{I} - \mathbf{S})$. To do so, solve the *consistent but underdetermined* system

$$\begin{bmatrix} \mathrm{i} & 1 \\ -1 & \mathrm{i} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

The first row requires

$$\mathrm{i}v_1 + v_2 = 0,$$

while the second row requires

$$-v_1 + \mathrm{i}v_2 = 0.$$

Multiply that last equation by $-i$ and you obtain the first equation: so if you satisfy the second equation ($v_1 = \mathrm{i}v_2$), you satisfy them both. Thus let

$$\mathbf{v} = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} \mathrm{i}v_2 \\ v_2 \end{bmatrix}.$$

The specific eigenvector $\mathbf{v}_1$ (given in the main text) follows from picking $v_2 = 1$.

Suppose we want to compute the norm of the eigenvector $\mathbf{v}_1$. Using our usual method, we would have

$$\|\mathbf{v}_1\|^2 = \mathbf{v}_1^T\mathbf{v}_1 = \begin{bmatrix} \mathrm{i} & 1 \end{bmatrix} \begin{bmatrix} \mathrm{i} \\ 1 \end{bmatrix} = \mathrm{i}^2 + 1 = -1 + 1 = 0.$$

This result seems strange, no? How could the norm of a nonzero vector be zero?

This example reveals a crucial shortcoming in our definition of the norm, when applied to complex vectors. Instead of

$$\|\mathbf{x}\| = \sqrt{x_1^2 + x_2^2 + \cdots + x_n^2} = \sqrt{\mathbf{x}^T\mathbf{x}},$$

we want

$$\|\mathbf{x}\| = \sqrt{|x_1|^2 + |x_2|^2 + \cdots + |x_n|^2}.$$

If $z = a + \mathrm{i}b \in \mathbb{C}$ with $a, b \in \mathbb{R}$, then $|z| = \sqrt{a^2 + b^2}$. We call $|z|$ then *magnitude* of the complex number z.

For real vectors $\mathbf{x} \in \mathbb{R}^n$, both definitions are the same. For complex vectors $\mathbf{x} \in \mathbb{C}^n$ they can be very different. Just as the norm of a real vector has the compact notation $\|\mathbf{v}\| = \sqrt{\mathbf{v}^T\mathbf{v}}$, so too does the norm of a complex vector:

$$\|\mathbf{v}\| = \sqrt{\overline{\mathbf{v}}^T\mathbf{v}},$$

where $\overline{\mathbf{v}}$ denotes the *complex conjugate* of $\mathbf{v}$. Now apply this definition of the norm to $\mathbf{v}_1$:

The *complex conjugate* of $z = a + \mathrm{i}b$ is

$$\overline{z} = a - \mathrm{i}b,$$

allowing us to write

$$\begin{aligned} |z|^2 = \overline{z}z &= (a - \mathrm{i}b)(a + \mathrm{i}) \\ &= a^2 - \mathrm{i}b + \mathrm{i}b + a^2 = a^2 + b^2. \end{aligned}$$

$$\|\mathbf{v}_1\|^2 = \overline{\mathbf{v}_1}^T\mathbf{v}_1 = \begin{bmatrix} -\mathrm{i} & 1 \end{bmatrix} \begin{bmatrix} \mathrm{i} \\ 1 \end{bmatrix} = -\mathrm{i}^2 + 1 = 1 + 1 = 2,$$

a much more reasonable answer than we had before.

We will wrap up this complex interlude by proving that the real symmetric matrices that will be our focus in this course can never have complex eigenvalues.

6.1.2 *The spectral theorem for symmetric matrices*

Theorem 8 *All eigenvalues of a real symmetric matrix are real.*

Proof. Let $\mathbf{S}$ denote a symmetric matrix with real entries, so $\mathbf{S}^T = \mathbf{S}$ (since $\mathbf{S}$ is symmetric) and $\overline{\mathbf{S}} = \mathbf{S}$ (since $\mathbf{S}$ is real).

Let $(\lambda, \mathbf{v})$ be an arbitrary eigenpair of $\mathbf{S}$, so that $\mathbf{S}\mathbf{v} = \lambda\mathbf{v}$. Without loss of generality, we can assume that $\mathbf{v}$ is scaled so that $\|\mathbf{v}\| = 1$, i.e., $\overline{\mathbf{v}}^T\mathbf{v} = \|\mathbf{v}\|^2 = 1$. Thus

$$\lambda = \lambda\|\mathbf{v}\|^2 = \lambda(\overline{\mathbf{v}}^T\mathbf{v}) = \overline{\mathbf{v}}^T(\lambda\mathbf{v}) = \overline{\mathbf{v}}^T(\mathbf{S}\mathbf{v}).$$

Since we do not yet know that $\mathbf{v}$ is real-valued, we must use the norm definition for complex vectors discussed in the previous subsection.

Since $\mathbf{S}$ is real and symmetric, $\mathbf{S} = \overline{\mathbf{S}}^T$, and so

$$\overline{\mathbf{v}}^T(\mathbf{S}\mathbf{v}) = \overline{\mathbf{v}}^T\overline{\mathbf{S}}^T\mathbf{v} = \overline{(\mathbf{S}\mathbf{v})}^T\mathbf{v} = \overline{(\lambda\mathbf{v})}^T\mathbf{v} = \overline{\lambda}\overline{\mathbf{v}}^T\mathbf{v} = \overline{\lambda}\|\mathbf{v}\|^2 = \overline{\lambda}.$$

If $z = a + \mathrm{i}b$ and $z = \overline{z}$, then $a + \mathrm{i}b = a - \mathrm{i}b$, i.e.,

$$b = -b,$$

which is only possible if $b = 0$.

We have shown that $\lambda = \overline{\lambda}$, which is only possible if λ is real. ∎

It immediately follows that if λ is an eigenvalue of the real symmetric matrix $\mathbf{S}$, then we can always find a real-valued eigenvector $\mathbf{v}$ of $\mathbf{S}$ corresponding to λ, simply by finding a real-valued vector in the null space

$$\mathcal{N}(\lambda\mathbf{I} - \mathbf{S}),$$

since $\lambda\mathbf{I} - \mathbf{S}$ is a real-valued matrix.

Crucially, the eigenvalues of a real symmetric matrix $\mathbf{S}$ associated with distinct eigenvalues must be orthogonal.

Theorem 9 *Eigenvectors of a real symmetric matrix associated with distinct eigenvalues are orthogonal.*

Proof. Suppose λ and γ are distinct eigenvalues of a real symmetric matrix $\mathbf{S}$ associated with eigenvectors $\mathbf{v} \in \mathbb{R}^n$ and $\mathbf{w} \in \mathbb{R}^n$:

$$\mathbf{S}\mathbf{v} = \lambda\mathbf{v}, \qquad \mathbf{S}\mathbf{w} = \gamma\mathbf{w}$$

with $\lambda \neq \gamma$. Now consider

$$\lambda\mathbf{w}^T\mathbf{v} = \mathbf{w}^T(\lambda\mathbf{v}) = \mathbf{w}^T(\mathbf{S}\mathbf{v}) = \mathbf{w}^T\mathbf{S}^T\mathbf{v},$$

where we have used the fact that $\mathbf{S} = \mathbf{S}^T$. Now

$$\mathbf{w}^T\mathbf{S}^T\mathbf{v} = (\mathbf{S}\mathbf{w})^T\mathbf{v} = (\gamma\mathbf{w})^T\mathbf{v} = \gamma\mathbf{w}^T\mathbf{v}.$$

We have thus shown that

$$\lambda\mathbf{w}^T\mathbf{v} = \gamma\mathbf{w}^T\mathbf{v}.$$

Since $\lambda \neq \gamma$, this statement can only be true if $\mathbf{w}^T\mathbf{v} = 0$, i.e., if $\mathbf{v}$ and $\mathbf{w}$ are orthogonal. ∎

What if the eigenvalues are not *distinct*? Consider the simple 2×2 identity matrix, $\mathbf{I}$. Any nonzero $\mathbf{x} \in \mathbb{R}^2$ is an eigenvector of $\mathbf{I}$ associated with the eigenvalue $\lambda = 1$, since

$$\mathbf{I}\mathbf{x} = 1\mathbf{x}.$$

Thus we have many eigenvectors that are not orthogonal. However, we can always find vectors, like

$$\mathbf{v}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \mathbf{v}_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

that are orthogonal.

We are ready to collect relevant facts in the *Spectral Theorem*.

Theorem 10 (Spectral Theorem) *Suppose $\mathbf{S} \in \mathbb{R}^{n\times n}$ is symmetric, $\mathbf{S}^T = \mathbf{S}$. Then there exist n (not necessarily distinct) eigenvalues $\lambda_1, \ldots, \lambda_n$ and corresponding unit-length eigenvectors $\mathbf{v}_1, \ldots, \mathbf{v}_n$ such that*

$$\mathbf{S}\mathbf{v}_j = \lambda_j \mathbf{v}_j.$$

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

$$\mathbb{R}^n = \text{span}\{\mathbf{v}_1, \ldots, \mathbf{v}_n\}$$

and $\mathbf{v}_j^T \mathbf{v}_k = 0$ when $j \neq k$, and $\mathbf{v}_j^T \mathbf{v}_j = \|\mathbf{v}_j\|^2 = 1$.

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

$$\mathbf{S} = \sum_{j=1}^{n} \lambda_j \mathbf{v}_j \mathbf{v}_j^T. \tag{6.1}$$

This equation expresses $\mathbf{S}$ as the sum of the special rank-1 matrices $\lambda_j \mathbf{v}_j \mathbf{v}_j^T$. The singular value decomposition will provide a similar way to tease apart a rectangular matrix.

For example, when

$$\mathbf{S} = \begin{bmatrix} 3 & -1 \\ -1 & 3 \end{bmatrix},$$

we have $\lambda_1 = 4$ and $\lambda_2 = 2$, with

$$\mathbf{v}_1 = \begin{bmatrix} \sqrt{2}/2 \\ -\sqrt{2}/2 \end{bmatrix}, \quad \mathbf{v}_2 = \begin{bmatrix} \sqrt{2}/2 \\ \sqrt{2}/2 \end{bmatrix}.$$

Note that these eigenvectors are unit vectors, and they are orthogonal. We can write

$$\begin{aligned}\mathbf{S} &= \lambda_1 \mathbf{v}_1 \mathbf{v}_1^T + +\lambda_2 \mathbf{v}_2 \mathbf{v}_2^T \\ &= 4 \begin{bmatrix} 1/2 & -1/2 \\ -1/2 & 1/2 \end{bmatrix} + 2 \begin{bmatrix} 1/2 & 1/2 \\ 1/2 & 1/2 \end{bmatrix}.\end{aligned}$$

Definition 21 *A symmetric matrix $\mathbf{S} \in \mathbb{R}^{n\times n}$ is* positive definite *provided $\mathbf{x}^T\mathbf{S}\mathbf{x} > 0$ for all nonzero $\mathbf{x} \in \mathbb{R}^n$; if $\mathbf{x}^T\mathbf{S}\mathbf{x} \geq 0$ for all $\mathbf{x} \in \mathbb{R}^n$, we say $\mathbf{S}$ is* positive semidefinite.

For the example above,

$$\begin{aligned}\mathbf{x}^T\mathbf{S}\mathbf{x} &= \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^T \begin{bmatrix} 3 & -1 \\ -1 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \\ &= 3x_1^2 - 2x_1x_2 + x_2^2 \\ &= 2(x_1 - x_2)^2 + (x_1 + x_2)^2.\end{aligned}$$

This last expression, the sum of squares, is clearly positive for all nonzero $\mathbf{x}$, so $\mathbf{S}$ is positive definite.

Theorem 11 *All eigenvalues of a symmetric positive definite matrix are positive; all eigenvalues of a symmetric positive semidefinite matrix are nonnegative.*

Proof. Let $(\lambda_j, \mathbf{v}_j)$ denote an eigenpair of the symmetric positive definite matrix $\mathbf{S} \in \mathbb{R}^{n\times n}$ with $\|\mathbf{v}_j\|^2 = \mathbf{v}_j^T\mathbf{v}_j = 1$. Since $\mathbf{S}$ is symmetric , λ_j must be real. We conclude that

$$\lambda_j = \lambda_j \mathbf{v}_j^T \mathbf{v}_j = \mathbf{v}_j^T(\lambda_j \mathbf{v}_j) = \mathbf{v}_j^T \mathbf{S}\mathbf{v}_j,$$

which must be positive since $\mathbf{S}$ is positive definite and $\mathbf{v}_j \neq \mathbf{0}$.

The proof for positive semidefinite matrices is the same, except we can only conclude that $\lambda_j = \mathbf{v}_j^T\mathbf{S}\mathbf{v}_j \geq 0$. ∎

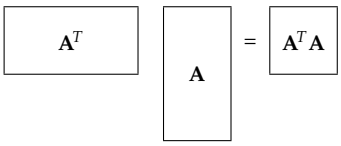
Can you prove the converse of this theorem? (A symmetric matrix with positive eigenvalues is positive definite.) Hint: use the Spectral Theorem. With this result, we can check if $\mathbf{S}$ is positive definite by just looking at its eigenvalues, rather than working out a formula for $\mathbf{x}^T\mathbf{S}\mathbf{x}$, as done above.

6.2 *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 $\mathbf{A} \in \mathbb{R}^{m\times n}$ with $m \geq n$, and that $\text{rank}(\mathbf{A})$ is as large as possible, i.e.,

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

First, form $\mathbf{A}^T\mathbf{A}$, which is an $n \times n$ matrix. Notice that $\mathbf{A}^T\mathbf{A}$ is always symmetric, since



$$(\mathbf{A}^T\mathbf{A})^T = \mathbf{A}^T(\mathbf{A}^T)^T = \mathbf{A}^T\mathbf{A}.$$

Furthermore, this matrix is positive definite: notice that

$$\mathbf{x}^T\mathbf{A}^T\mathbf{A}\mathbf{x} = (\mathbf{A}\mathbf{x})^T(\mathbf{A}\mathbf{x}) = \|\mathbf{A}\mathbf{x}\|^2 \geq 0.$$

Since $\text{rank}(\mathbf{A}) = n$, notice that

$$\dim(\mathcal{N}(\mathbf{A})) = n - \text{rank}(\mathbf{A}) = 0.$$

Keep this in mind: If $\text{rank}(\mathbf{A}) < n$, then $\dim(\mathcal{N}(\mathbf{A})) > 0$, so there exist $\mathbf{x} \neq \mathbf{0}$ for which $\mathbf{x}^T\mathbf{A}^T\mathbf{A}\mathbf{x} = \|\mathbf{A}\mathbf{x}\|^2 = 0$. Hence $\mathbf{A}^T\mathbf{A}$ will only be positive semidefinite in this case.

Since the null space of $\mathbf{A}$ is trivial, $\mathbf{A}\mathbf{x} \neq \mathbf{0}$ whenever $\mathbf{x} \neq \mathbf{0}$, so

$$\mathbf{x}^T\mathbf{A}^T\mathbf{A}\mathbf{x} = \|\mathbf{A}\mathbf{x}\|^2 > 0$$

for all nonzero $\mathbf{x}$. Hence $\mathbf{A}^T\mathbf{A}$ 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 $\mathbf{A}^T\mathbf{A}$.**

As a consequence of results about symmetric matrices presented above, we can find n eigenpairs $\{(\lambda_j, \mathbf{v}_j)\}_{j=1}^n$ of $\mathbf{S} = \mathbf{A}^T\mathbf{A}$ with unit eigenvectors ($\mathbf{v}_j^T\mathbf{v}_j = \|\mathbf{v}_j\|^2 = 1$) that are orthogonal to one another ($\mathbf{v}_j^T\mathbf{v}_k = 0$ 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 $\mathbf{v}_1, \ldots, \mathbf{v}_n \in \mathbb{R}^n$.

Even if $\mathbf{A}$ is a square matrix, be sure to compute the eigenvalues and eigenvectors of $\mathbf{A}^T\mathbf{A}$.

Since $\mathbf{A}^T\mathbf{A}$ is positive definite, all its eigenvalues are positive.

Step 2. **Define $\sigma_j = \|\mathbf{A}\mathbf{v}_j\| = \sqrt{\lambda_j}$, $j = 1, \ldots, n$.**

Note that $\sigma_j^2 = \|\mathbf{A}\mathbf{v}_j\|_2^2 = \mathbf{v}_j^T\mathbf{A}^T\mathbf{A}\mathbf{v}_j = \lambda_j$. Since the eigenvalues $\lambda_1, \ldots, \lambda_n$ are decreasing in size, so too are the σ_j values:

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n > 0.$$

Step 3. **Define $\mathbf{u}_j = \mathbf{A}\mathbf{v}_j/\sigma_j$ for $j = 1, \ldots, n$.**

Notice that $\mathbf{u}_1, \ldots, \mathbf{u}_n \in \mathbb{R}^m$. Because $\sigma_j = \|\mathbf{A}\mathbf{v}_j\|$, we ensure that

The assumption that $\text{rank}(\mathbf{A}) = n$ helped us out here, by ensuring that $\sigma_j > 0$ for all j: hence we can safely divide by σ_j in the definition of $\mathbf{u}_j$.

$$\|\mathbf{u}_j\| = \left\|\frac{1}{\sigma_j}\mathbf{A}\mathbf{v}_j\right\| = \frac{\|\mathbf{A}\mathbf{v}_j\|}{\sigma_j} = 1.$$

Furthermore, these $\mathbf{u}_j$ vectors are orthogonal. To see this, write

$$\mathbf{u}_j^T\mathbf{u}_k = \frac{1}{\sigma_j\sigma_k}(\mathbf{A}\mathbf{v}_j)^T(\mathbf{A}\mathbf{v}_k) = \frac{1}{\sigma_j\sigma_k}\mathbf{v}_j^T\mathbf{A}^T\mathbf{A}\mathbf{v}_k.$$

Since $\mathbf{v}_k$ is an eigenvector of $\mathbf{A}^T\mathbf{A}$ corresponding to eigenvalue λ_k,

$$\frac{1}{\sigma_j\sigma_k}\mathbf{v}_j^T\mathbf{A}^T\mathbf{A}\mathbf{v}_k = \frac{1}{\sigma_j\sigma_k}\mathbf{v}_j^T(\lambda_k\mathbf{v}_k) = \frac{\lambda_j}{\sigma_j\sigma_k}\mathbf{v}_j^T\mathbf{v}_k.$$

Since the eigenvectors of the symmetric matrix $\mathbf{A}^T\mathbf{A}$ are orthogonal, $\mathbf{v}_j^T\mathbf{v}_k = 0$ when $j \neq k$, so the $\mathbf{u}_j$ vectors inherit the orthogonality of the $\mathbf{v}_j$ vectors:

$$\mathbf{u}_j^T\mathbf{u}_k = 0, \qquad j \neq k.$$

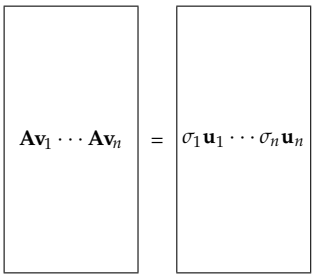
Step 4. **Put the pieces together.**

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

$$\mathbf{A}\mathbf{v}_j = \sigma_j\mathbf{u}_j,$$

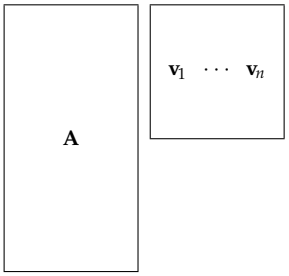
regardless of whether $\sigma_j = 0$ or not. We can stack these n vector equations as columns of a single matrix equation,

$$\begin{bmatrix} | & | & & | \\ \mathbf{A}\mathbf{v}_1 & \mathbf{A}\mathbf{v}_2 & \cdots & \mathbf{A}\mathbf{v}_n \\ | & | & & | \end{bmatrix} = \begin{bmatrix} | & | & & | \\ \sigma_1\mathbf{u}_1 & \sigma_2\mathbf{u}_2 & \cdots & \sigma_n\mathbf{u}_n \\ | & | & & | \end{bmatrix}.$$



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

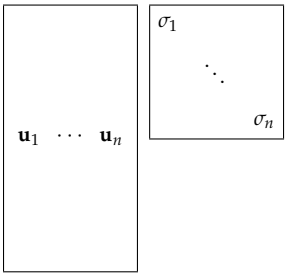
$$\mathbf{A}\begin{bmatrix} | & | & & | \\ \mathbf{v}_1 & \mathbf{v}_2 & \cdots & \mathbf{v}_n \\ | & | & & | \end{bmatrix} = \begin{bmatrix} | & | & & | \\ \mathbf{u}_1 & \mathbf{u}_2 & \cdots & \mathbf{u}_n \\ | & | & & | \end{bmatrix}\begin{bmatrix} \sigma_1 & & & \\ & \sigma_2 & & \\ & & \ddots & \\ & & & \sigma_n \end{bmatrix}.$$



Denote these matrices as

$$\mathbf{A}\mathbf{V} = \widehat{\mathbf{U}}\widehat{\mathbf{\Sigma}}, \tag{6.2}$$

where $\mathbf{A} \in \mathbb{R}^{m\times n}$, $\mathbf{V} \in \mathbb{R}^{n\times n}$, $\widehat{\mathbf{U}} \in \mathbb{R}^{m\times n}$, and $\widehat{\mathbf{\Sigma}} \in \mathbb{R}^{n\times n}$.



We now have all the ingredients for various forms of the singular value decomposition. Since the eigenvectors $\mathbf{v}_j$ of the symmetric matrix $\mathbf{A}^T\mathbf{A}$ are orthonormal, the square matrix $\mathbf{V}$ has orthonormal columns. This means that

$$\mathbf{V}^T\mathbf{V} = \mathbf{I},$$

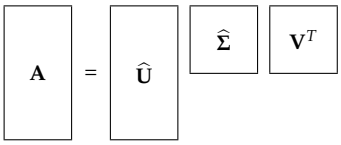
since the (j,k) entry of $\mathbf{V}^T\mathbf{V}$ is simply $\mathbf{v}_j^T\mathbf{v}_k$. Since $\mathbf{V}$ is square, the equation $\mathbf{V}^T\mathbf{V} = \mathbf{I}$ implies that $\mathbf{V}^T = \mathbf{V}^{-1}$. Thus, in addition to $\mathbf{V}^T\mathbf{V}$, we also have

$$\mathbf{V}\mathbf{V}^T = \mathbf{V}\mathbf{V}^{-1} = \mathbf{I}.$$

The inverse of a square matrix is unique: since $\mathbf{V}^T$ does what the inverse of $\mathbf{V}$ is supposed to do, i.e., $\mathbf{V}^T\mathbf{V} = \mathbf{I}$, it must be the unique matrix $\mathbf{V}^{-1}$.

Thus multiplying both sides of equation (6.2) on the right by $\mathbf{V}^T$ gives

$$\mathbf{A} = \widehat{\mathbf{U}}\widehat{\mathbf{\Sigma}}\mathbf{V}^T. \tag{6.3}$$



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

```
[Uhat, Sighat, V] = svd(A,0).
```

What can be said of the matrix $\widehat{\mathbf{U}} \in \mathbb{R}^{m\times n}$? Recall that its columns, the vectors $\mathbf{u}_1, \ldots, \mathbf{u}_n$, are orthonormal. However, in contrast to $\mathbf{V}$, *we cannot conclude that* $\widehat{\mathbf{U}}\widehat{\mathbf{U}}^T = \mathbf{I}$ *when* $m > n$. Why not? Because when $m > n$, $\widehat{\mathbf{U}}^T \in \mathbb{R}^{n\times m}$ has a nontrivial null space, and hence cannot be invertible.

When $m > n$, there must exist some nonzero $\mathbf{z} \in \mathbb{R}^m$ such that $\mathbf{z} \perp \mathbf{u}_1, \ldots, \mathbf{u}_n$, which implies $\widehat{\mathbf{U}}^T\mathbf{z} = \mathbf{0}$. Hence $\widehat{\mathbf{U}}\widehat{\mathbf{U}}^T\mathbf{z} = \mathbf{0}$, so we cannot have $\widehat{\mathbf{U}}\widehat{\mathbf{U}}^T = \mathbf{I}$. However, $\widehat{\mathbf{U}}\widehat{\mathbf{U}}^T \in \mathbb{R}^{m\times m}$ is a projector onto the n-dimensional subspace $\text{span}\{\mathbf{u}_1, \ldots, \mathbf{u}_n\}$ of $\mathbb{R}^m$.

We wish to augment the matrix $\widehat{\mathbf{U}}$ with $m - n$ additional column vectors, to give a full set of m orthonormal vectors in $\mathbb{R}^m$. Here is the recipe to find these extra vectors: For $j = n+1, \ldots, m$, pick

$$\mathbf{u}_j \perp \text{span}\{\mathbf{u}_1, \ldots, \mathbf{u}_{j-1}\}$$

with $\mathbf{u}_j^T\mathbf{u}_j = 1$. Then define

$$\mathbf{U} = \left[\begin{array}{ccccccc} | & & | & | & & | \\ \mathbf{u}_1 & \cdots & \mathbf{u}_n & \mathbf{u}_{n+1} & \cdots & \mathbf{u}_m \\ | & & | & | & & | \end{array}\right] \in \mathbb{R}^{m\times m}. \tag{6.4}$$

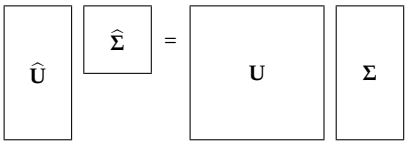
We have constructed $\mathbf{u}_1, \ldots, \mathbf{u}_m$ to be orthonormal vectors, so

$$\mathbf{U}^T\mathbf{U} = \mathbf{I}.$$

However, since $\mathbf{U} \in \mathbb{R}^{m\times m}$, this orthogonality also implies $\mathbf{U}^{-1} = \mathbf{U}^T$.

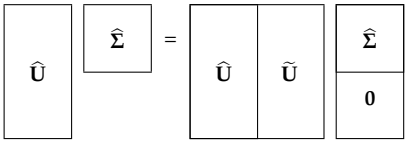
Now we are ready to replace the rectangular matrix $\widehat{\mathbf{U}} \in \mathbb{R}^{m\times n}$ in the reduced SVD (6.3) with the square matrix $\mathbf{U} \in \mathbb{R}^{m\times m}$. To do so, we also need to replace $\widehat{\mathbf{\Sigma}} \in \mathbb{R}^{n\times n}$ by some $\mathbf{\Sigma} \in \mathbb{R}^{m\times n}$ in such a way that

$$\widehat{\mathbf{U}}\widehat{\mathbf{\Sigma}} = \mathbf{U}\mathbf{\Sigma}.$$

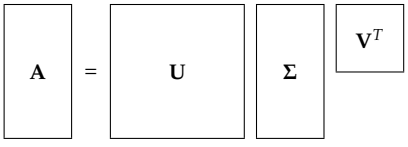


The simplest approach is to obtain $\mathbf{\Sigma}$ by appending zeros to the end of $\widehat{\mathbf{\Sigma}}$, thus ensuring there is no contribution when the new entries of $\mathbf{U}$ multiply against the new entries of $\mathbf{\Sigma}$:

$$\mathbf{\Sigma} = \left[\begin{array}{c} \widehat{\mathbf{\Sigma}} \\ \mathbf{0} \end{array}\right] \in \mathbb{R}^{m\times n}. \tag{6.5}$$



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



Theorem 12 (Singular value decomposition, provisional version) *Suppose $\mathbf{A} \in \mathbb{R}^{m\times n}$ has* $\operatorname{rank}(\mathbf{A}) = n$*, with $m \geq n$. Then we can write*

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

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

$$\mathbf{U}^T\mathbf{U} = \mathbf{I} \in \mathbb{R}^{m\times m}, \qquad \mathbf{V}^T\mathbf{V} = \mathbf{I} \in \mathbb{R}^{n\times n},$$

and $\boldsymbol{\Sigma} \in \mathbb{R}^{m\times n}$ is zero everywhere except for entries on the main diagonal, where the (j, j) entry is σ_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] = svd(A).
```

Definition 22 *Let $\mathbf{A} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$ be a full singular value decomposition. The diagonal entries of $\boldsymbol{\Sigma}$, denoted $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n$, are called the* singular values *of $\mathbf{A}$. The columns $\mathbf{u}_1, \ldots, \mathbf{u}_m$ of $\mathbf{U}$ are the* left singular vectors*; the columns $\mathbf{v}_1, \ldots, \mathbf{v}_m$ of $\mathbf{V}$ are the* right singular vectors*.*

6.3 The dyadic form of the SVD

We are now prepared to develop an analogue of the formula (6.1) for rectangular matrices. Consider the reduced SVD,

$$\mathbf{A} = \widehat{\mathbf{U}}\widehat{\boldsymbol{\Sigma}}\mathbf{V}^T,$$

and multiply $\widehat{\mathbf{U}}\widehat{\boldsymbol{\Sigma}}$ to obtain

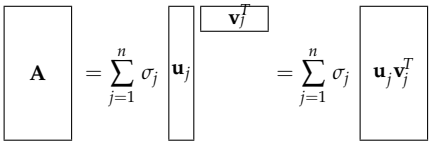
$$\begin{bmatrix} | & | & & | \\ \mathbf{u}_1 & \mathbf{u}_2 & \cdots & \mathbf{u}_n \\ | & | & & | \end{bmatrix} \begin{bmatrix} \sigma_1 & & & \\ & \sigma_2 & & \\ & & \ddots & \\ & & & \sigma_n \end{bmatrix} = \begin{bmatrix} | & | & & | \\ \sigma_1\mathbf{u}_1 & \sigma_1\mathbf{u}_2 & \cdots & \sigma_n\mathbf{u}_n \\ | & | & & | \end{bmatrix}.$$

Now notice that you can write $\mathbf{A} = (\widehat{\mathbf{U}}\widehat{\boldsymbol{\Sigma}})\mathbf{V}^T$ as

$$\begin{bmatrix} | & | & & | \\ \sigma_1\mathbf{u}_1 & \sigma_1\mathbf{u}_2 & \cdots & \sigma_n\mathbf{u}_n \\ | & | & & | \end{bmatrix} \begin{bmatrix} — & \mathbf{v}_1^T & — \\ — & \mathbf{v}_2^T & — \\ & \vdots & \\ — & \mathbf{v}_n^T & — \end{bmatrix} = \sum_{j=1}^{n} \sigma_j \mathbf{u}_j \mathbf{v}_j^T,$$

which parallels the form (6.1) we had for symmetric matrices:

$$\mathbf{A} = \sum_{j=1}^{n} \sigma_j \mathbf{u}_j \mathbf{v}_j^T. \tag{6.6}$$



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.

6.4 A small example

Consider the matrix

$$\mathbf{A} = \begin{bmatrix} 1 & 1 \\ 0 & 0 \\ \sqrt{2} & -\sqrt{2} \end{bmatrix},$$

for which $\mathbf{A}^T\mathbf{A}$ is the symmetric matrix used as an example earlier in these notes:

$$\mathbf{A}^T\mathbf{A} = \begin{bmatrix} 3 & -1 \\ -1 & 3 \end{bmatrix}.$$

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

Step 1. **Compute the eigenvalues and eigenvectors of $\mathbf{A}^T\mathbf{A}$.**

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

$$\mathbf{v}_1 = \begin{bmatrix} \sqrt{2}/2 \\ -\sqrt{2}/2 \end{bmatrix}, \quad \mathbf{v}_2 = \begin{bmatrix} \sqrt{2}/2 \\ \sqrt{2}/2 \end{bmatrix},$$

with $\lambda_1 \geq \lambda_2$, the required order. The vectors $\mathbf{v}_1$ and $\mathbf{v}_2$ will be the right singular vectors of $\mathbf{A}$.

Step 2. **Define $\sigma_j = \|\mathbf{A}\mathbf{v}_j\| = \sqrt{\lambda_j}$, $j = 1, \ldots, n$.**

In this case, we compute

$$\sigma_1 = \sqrt{\lambda_1} = 2, \qquad \sigma_2 = \sqrt{\lambda_2} = \sqrt{2}.$$

Alternatively, we could have computed the singular values from

$$\mathbf{A}\mathbf{v}_1 = \begin{bmatrix} 1 & 1 \\ 0 & 0 \\ \sqrt{2} & -\sqrt{2} \end{bmatrix} \begin{bmatrix} \sqrt{2}/2 \\ -\sqrt{2}/2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 2 \end{bmatrix}$$

$$\mathbf{A}\mathbf{v}_2 = \begin{bmatrix} 1 & 1 \\ 0 & 0 \\ \sqrt{2} & -\sqrt{2} \end{bmatrix} \begin{bmatrix} \sqrt{2}/2 \\ \sqrt{2}/2 \end{bmatrix} = \begin{bmatrix} \sqrt{2} \\ 0 \\ 0 \end{bmatrix},$$

with $\sigma_1 = \|\mathbf{A}\mathbf{v}_1\| = 2$ and $\sigma_2 = \|\mathbf{A}\mathbf{v}_2\| = \sqrt{2}$.

Step 3. **Define $\mathbf{u}_j = \mathbf{A}\mathbf{v}_j/\sigma_j$, $j = 1, \ldots, n$.**

We use the vectors $\mathbf{A}\mathbf{v}_1$ and $\mathbf{A}\mathbf{v}_2$ computed at the last step:

$$\mathbf{u}_1 = \frac{1}{\sigma_1}\mathbf{A}\mathbf{v}_1 = \frac{1}{2}\begin{bmatrix} 0 \\ 0 \\ 2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \qquad \mathbf{u}_2 = \frac{1}{\sigma_2}\mathbf{A}\mathbf{v}_2 = \frac{1}{\sqrt{2}}\begin{bmatrix} \sqrt{2} \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}.$$

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

We immediately have the *reduced SVD* $\mathbf{A} = \widehat{\mathbf{U}}\widehat{\mathbf{\Sigma}}\mathbf{V}^T$:

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

To get the full SVD, we need a unit vector $\mathbf{u}_3$ that is orthogonal to $\mathbf{u}_1$ and $\mathbf{u}_2$. In this case, such a vector is easy to spot:

$$\mathbf{u}_3 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}.$$

Thus we can write the *full SVD* $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$:

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

Finally, we write the *dyadic form of the SVD*, $\mathbf{A} = \sum_{j=1}^{2} \sigma_j \mathbf{u}_j \mathbf{v}_j^T$:

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

6.5 *Derivation of the singular value decomposition: Rank deficient case*

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

Recall that the dimension of the null space of $\mathbf{A}$ is given by

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

How do the null spaces of $\mathbf{A}$ and $\mathbf{A}^T\mathbf{A}$ compare?

Lemma 1 *For any matrix* $\mathbf{A} \in \mathbb{R}^{m \times n}$, $\mathcal{N}(\mathbf{A}^T\mathbf{A}) = \mathcal{N}(\mathbf{A})$.

Proof. First we show that $\mathcal{N}(\mathbf{A})$ is contained in $\mathcal{N}(\mathbf{A}^T\mathbf{A})$. If $\mathbf{x} \in \mathcal{N}(\mathbf{A})$, then $\mathbf{A}\mathbf{x} = \mathbf{0}$. Premultiplying by $\mathbf{A}^T$ gives $\mathbf{A}^T\mathbf{A}\mathbf{x} = \mathbf{0}$, so $\mathbf{x} \in \mathcal{N}(\mathbf{A}^T\mathbf{A})$.

Now we show that $\mathcal{N}(\mathbf{A}^T\mathbf{A})$ is contained in $\mathcal{N}(\mathbf{A})$. If $\mathbf{x} \in \mathcal{N}(\mathbf{A}^T\mathbf{A})$, then $\mathbf{A}^T\mathbf{A}\mathbf{x} = \mathbf{0}$. Premultiplying by $\mathbf{x}^T$ gives

$$\mathbf{0} = \mathbf{x}^T\mathbf{A}^T\mathbf{A}\mathbf{x} = (\mathbf{A}\mathbf{x})^T(\mathbf{A}\mathbf{x}) = \|\mathbf{A}\mathbf{x}\|^2.$$

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

Since the spaces $\mathcal{N}(\mathbf{A})$ and $\mathcal{N}(\mathbf{A}^T\mathbf{A})$ each contain the other, we conclude that $\mathcal{N}(\mathbf{A}) = \mathcal{N}(\mathbf{A}^T\mathbf{A})$. ∎

Now we can make a crucial insight: the dimension of $\mathcal{N}(\mathbf{A})$ tells us how many zero eigenvalues $\mathbf{A}^T\mathbf{A}$ has. In particular, suppose $\mathbf{x}_1, \ldots, \mathbf{x}_{n-r}$ is a basis for $\mathcal{N}(\mathbf{A})$. Then $\mathbf{Ax}_j = \mathbf{0}$ implies

$$\begin{aligned} \mathbf{A}^T\mathbf{A}\mathbf{x}_j &= \mathbf{0}, \qquad j = 1, \ldots, n-r \\ &= 0\mathbf{x}_j, \end{aligned}$$

and so $\lambda = 0$ is an eigenvalue of $\mathbf{A}^T\mathbf{A}$ of multiplicity $n - r$.

Can you construct a 2×2 matrix $\mathbf{A}$ whose only eigenvalue is zero, but $\dim(\mathcal{N}(\mathbf{A})) = 1$? What is the multiplicity of the zero eigenvalue of $\mathbf{A}^T\mathbf{A}$?

How do these zero eigenvalues of $\mathbf{A}^T\mathbf{A}$ 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 $\mathbf{A}^T\mathbf{A}$.**

Since we order the eigenvalues of $\mathbf{A}^T\mathbf{A}$ so that $\lambda_1 \geq \cdots \geq \lambda_n \geq 0$, and we have just seen that zero is an eigenvalue of $\mathbf{A}^T\mathbf{A}$ of multiplicity $n - r$, we must have

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_r > 0, \qquad \lambda_{r+1} = \cdots = \lambda_n = 0.$$

The corresponding orthonormal eigenvectors are $\mathbf{v}_1, \ldots, \mathbf{v}_n$, with the last $n - r$ of these vectors in $\mathcal{N}(\mathbf{A}^T\mathbf{A}) = \mathcal{N}(\mathbf{A})$, i.e., $\mathbf{Av}_j = \mathbf{0}$.

Step 2. **Define $\sigma_j = \|\mathbf{Av}_j\| = \sqrt{\lambda_j}$, $j = 1, \ldots, n$.**

This step proceeds without any alterations, though now we have

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0, \qquad \sigma_{r+1} = \cdots = \sigma_n = 0.$$

The third step of the SVD construction needs alteration, since we can only define the left singular vectors via $\mathbf{u}_j = \mathbf{Av}_j/\sigma_j$ when $\sigma_j > 0$, that is, for $j = 1, \ldots, r$. Any choice for the remaining vectors, $\mathbf{u}_{r+1}, \ldots, \mathbf{u}_n$, will trivially satisfy the equation $\mathbf{Av}_j = \sigma_j\mathbf{u}_j$, since $\mathbf{Av}_j = \mathbf{0}$ and $\sigma_j = 0$ for $j = r+1, \ldots, n$. Since we are building $\widehat{\mathbf{U}} \in \mathbb{R}^{m \times n}$ (and eventually $\mathbf{U} \in \mathbb{R}^{m \times m}$) to have orthonormal, we will simply build out $\mathbf{u}_{r+1}, \ldots, \mathbf{u}_n$ so that all the vectors $\mathbf{u}_1, \ldots, \mathbf{u}_n$ are orthonormal.

Step 3a. **Define $\mathbf{u}_j = \mathbf{Av}_j/\sigma_j$ for $j = 1, \ldots, r$.**

Step 3b. **Construct orthonormal vectors $\mathbf{u}_{r+1}, \ldots, \mathbf{u}_n$.**

For each $j = r+1, \ldots, n$, construct a *unit vector* $\mathbf{u}_j$ such that

$$\mathbf{u}_j \perp \operatorname{span}\{\mathbf{u}_1, \ldots, \mathbf{u}_{j-1}\}.$$

If $r = 0$ (which implies the trivial case $\mathbf{A} = \mathbf{0}$), just set $\mathbf{u}_1$ to be any unit vector.

This procedure is exactly the same as used above to construct the vectors $\mathbf{u}_{n+1}, \ldots, \mathbf{u}_m$ to extend the reduced SVD with $\widehat{\mathbf{U}} \in \mathbb{R}^{m \times n}$ to the full SVD with $\mathbf{U} \in \mathbb{R}^{m \times m}$.

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

This step proceeds exactly as before. Now we define

$$\widehat{\mathbf{U}} = \begin{bmatrix} | & & | & | & & | \\ \mathbf{u}_1 & \cdots & \mathbf{u}_r & \mathbf{u}_{r+1} & \cdots & \mathbf{u}_n \\ | & & | & | & & | \end{bmatrix} \in \mathbb{R}^{m\times n},$$

$$\widehat{\mathbf{\Sigma}} = \begin{bmatrix} \sigma_1 & & & & & \\ & \ddots & & & & \\ & & \sigma_r & & & \\ & & & \sigma_{r+1} & & \\ & & & & \ddots & \\ & & & & & \sigma_n \end{bmatrix} = \begin{bmatrix} \sigma_1 & & & & & \\ & \ddots & & & & \\ & & \sigma_r & & & \\ & & & 0 & & \\ & & & & \ddots & \\ & & & & & 0 \end{bmatrix} \in \mathbb{R}^{n\times n},$$

and

$$\mathbf{V} = \begin{bmatrix} | & & | & | & & | \\ \mathbf{v}_1 & \cdots & \mathbf{v}_r & \mathbf{v}_{r+1} & \cdots & \mathbf{v}_n \\ | & & | & | & & | \end{bmatrix} \in \mathbb{R}^{n\times n}.$$

Notice that $\mathbf{V}$ is still a square matrix with orthonormal columns, so $\mathbf{V}^T\mathbf{V} = \mathbf{I}$ and $\mathbf{V}^{-1} = \mathbf{V}^T$. Since $\mathbf{A}\mathbf{v}_j = \sigma_j\mathbf{u}_j$ holds for $j = 1,\ldots,n$, we again have the reduced singular value decomposition

$$\mathbf{A} = \widehat{\mathbf{U}}\widehat{\mathbf{\Sigma}}\mathbf{V}^T.$$

As before, $\widehat{\mathbf{U}} \in \mathbb{R}^{m\times n}$ can be enlarged to give $\mathbf{U} \in \mathbb{R}^{n\times n}$ by supplying extra orthogonal unit vectors that complete a basis for $\mathbb{R}^m$:

$$\mathbf{u}_j \perp \operatorname{span}\{\mathbf{u}_1,\ldots,\mathbf{u}_{j-1}\}, \quad \|\mathbf{u}_j\| = 1, \quad j = n+1,\ldots,m.$$

Constructing $\mathbf{U} \in \mathbb{R}^{m\times m}$ as in (6.4) and $\Sigma \in \mathbb{R}^{m\times n}$ as in (6.5), we have the full singular value decomposition

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T.$$

The dyadic decomposition could still be written as

$$\mathbf{A} = \sum_{j=1}^{n} \sigma_j\mathbf{u}_j\mathbf{v}_j^T,$$

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:

$$\mathbf{A} = \sum_{j=1}^{r} \sigma_j\mathbf{u}_j\mathbf{v}_j^T.$$

We will see that this form of $\mathbf{A}$ is especially useful for understanding the four fundamental subspaces.

6.6 The connection to $\mathbf{A}\mathbf{A}^T$

Our derivation of the SVD relied heavily on an eigenvalue decomposition of $\mathbf{A}^T\mathbf{A}$. How does the SVD relate to $\mathbf{A}\mathbf{A}^T$? Consider forming

$$\begin{aligned}\mathbf{A}\mathbf{A}^T &= (\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T)(\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T)^T \\ &= \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T\mathbf{V}\mathbf{\Sigma}^T\mathbf{U}^T \\ &= \mathbf{U}\mathbf{\Sigma}\mathbf{\Sigma}^T\mathbf{U}^T. \end{aligned} \tag{6.7}$$

Notice that $\mathbf{\Sigma}\mathbf{\Sigma}^T$ is a diagonal $m \times m$ matrix:

$$\mathbf{\Sigma}\mathbf{\Sigma}^T = \begin{bmatrix} \widehat{\mathbf{\Sigma}} \\ \mathbf{0} \end{bmatrix} \begin{bmatrix} \widehat{\mathbf{\Sigma}}^T & \mathbf{0} \end{bmatrix} = \begin{bmatrix} \widehat{\mathbf{\Sigma}}^2 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix},$$

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

$$\widehat{\mathbf{\Sigma}}^2 = \begin{bmatrix} \sigma_1^2 & & \\ & \ddots & \\ & & \sigma_n^2 \end{bmatrix} = \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{bmatrix},$$

where the λ_j values still denote the eigenvalues of $\mathbf{A}^T\mathbf{A}$. Thus equation (6.7) becomes

$$\mathbf{A}\mathbf{A}^T = \mathbf{U} \begin{bmatrix} \Lambda & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \mathbf{U}^T,$$

which is a diagonalization of $\mathbf{A}\mathbf{A}^T$. Postmultiplying this equation by $\mathbf{U}$, we have

$$(\mathbf{A}\mathbf{A}^T)\mathbf{U} = \mathbf{U} \begin{bmatrix} \Lambda & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix};$$

the first n columns of this equation give

$$\mathbf{A}\mathbf{A}^T\mathbf{u}_j = \lambda_j\mathbf{u}_j, \qquad j = 1, \ldots, n,$$

while the last $m - n$ columns give

$$\mathbf{A}\mathbf{A}^T\mathbf{u}_j = 0\mathbf{u}_j, \qquad j = n+1, \ldots, m.$$

Thus the columns $\mathbf{u}_1, \ldots, \mathbf{u}_n$ are eigenvectors of $\mathbf{A}\mathbf{A}^T$. Notice then that $\mathbf{A}\mathbf{A}^T$ and $\mathbf{A}^T\mathbf{A}$ have the same eigenvalues, except that $\mathbf{A}\mathbf{A}^T$ has $m - n$ extra zero eigenvalues.

This suggests a different way to compute the $\mathbf{U}$ matrix: form $\mathbf{A}\mathbf{A}^T$ and compute all its eigenvectors, giving $\mathbf{u}_1, \ldots, \mathbf{u}_m$ all at once. Thus we avoid the need for a special procedure to construct unit vectors orthogonal to $\mathbf{u}_1, \ldots, \mathbf{u}_r$.

6.7 Modification for the case of $m < n$

How does the singular value decomposition change if $\mathbf{A}$ has more columns than rows, $n > m$? The answer is easy: write the SVD of $\mathbf{A}^T$ (which has more rows than columns) using the procedure above, then take the 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 $r = \text{rank}(\mathbf{A}) \leq m$.

Step 1. **Compute the eigenvalues and eigenvectors of $\mathbf{A}\mathbf{A}^T$.**

Label the eigenvalues of $\mathbf{A}\mathbf{A}^T \in \mathbb{R}^{m\times m}$ as

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

and corresponding orthonormal eigenvectors as

$$\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_m$$

Step 2. **Define $\sigma_j = \|\mathbf{A}^T\mathbf{u}_j\| = \sqrt{\lambda_j}$, $j = 1, \ldots, m$.**

Step 3a. **Define $\mathbf{v}_j = \mathbf{A}^T\mathbf{u}_j/\sigma_j$ for $j = 1, \ldots, r$.**

Step 3b. **Construct orthonormal vectors $\mathbf{v}_{r+1}, \ldots, \mathbf{v}_m$.**

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

Steps 3a and 3b construct a matrix $\widehat{\mathbf{V}} \in \mathbb{R}^{n\times m}$ with orthonormal columns.

Step 3c. **Construct orthonormal vectors $\mathbf{v}_{m+1}, \ldots, \mathbf{v}_n$.**

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

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

First, defining

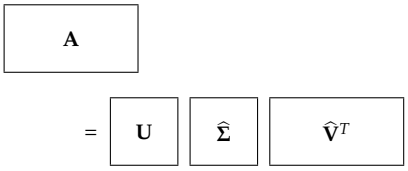
$$\widehat{\mathbf{U}} = \begin{bmatrix} | & & | \\ \mathbf{u}_1 & \cdots & \mathbf{u}_m \\ | & & | \end{bmatrix} \in \mathbb{R}^{m\times m}, \qquad \widehat{\mathbf{V}} = \begin{bmatrix} | & & | \\ \mathbf{v}_1 & \cdots & \mathbf{v}_m \\ | & & | \end{bmatrix} \in \mathbb{R}^{n\times m},$$

with diagonal matrix

$$\widehat{\mathbf{\Sigma}} = \text{diag}(\sigma_1, \ldots, \sigma_m) \in \mathbb{R}^{m\times m},$$

we have the reduced SVD

$$\mathbf{A} = \mathbf{U}\widehat{\mathbf{\Sigma}}\widehat{\mathbf{V}}^T.$$



To obtain the full SVD, we extend $\widehat{\mathbf{V}}$ to obtain

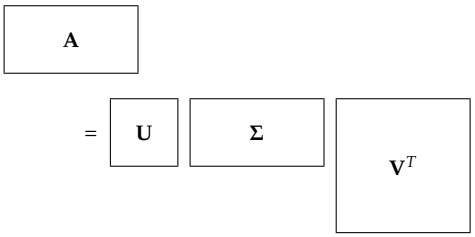
$$\mathbf{V} = \begin{bmatrix} | & & | & | & & | \\ \mathbf{v}_1 & \cdots & \mathbf{v}_m & \mathbf{v}_{m+1} & \cdots & \mathbf{v}_n \\ | & & | & | & & | \end{bmatrix} \in \mathbb{R}^{n\times n},$$

and similarly extend $\widehat{\mathbf{\Sigma}}$,

$$\mathbf{\Sigma} = \begin{bmatrix} \widehat{\mathbf{\Sigma}} & \mathbf{0} \end{bmatrix} \in \mathbb{R}^{m\times n}.$$

where we have now added extra zero columns, in contrast to the extra zero rows added in the $m > n$ case in (6.5). We thus arrive at the full SVD,

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T.$$



6.8 General statement of the singular value decomposition

We now can state the singular value decomposition in its fullest generality.

Theorem 13 (Singular value decomposition) *Suppose* $\mathbf{A} \in \mathbb{R}^{m\times n}$ *has* $\text{rank}(\mathbf{A}) = r$. *Then we can write*

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T,$$

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

$$\mathbf{U}^T\mathbf{U} = \mathbf{I} \in \mathbb{R}^{m\times m}, \qquad \mathbf{V}^T\mathbf{V} = \mathbf{I} \in \mathbb{R}^{n\times n},$$

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

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

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

Denoting the columns of $\mathbf{U}$ *and* $\mathbf{V}$ *as* $\mathbf{u}_1, \ldots, \mathbf{u}_m$ *and* $\mathbf{v}_1, \ldots, \mathbf{v}_m$, *we can write*

$$\mathbf{A} = \sum_{j=1}^{r} \sigma_j \mathbf{u}_j \mathbf{v}_j^T. \tag{6.8}$$

6.9 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

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

the row space

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

the null space

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

and the left null space

$$\mathcal{N}(\mathbf{A}^T) = \{\mathbf{y} \in \mathbb{R}^m : \mathbf{A}^T\mathbf{y} = \mathbf{0}\} \subseteq \mathbb{R}^m.$$

We shall explore these spaces using the dyadic form of the SVD (6.8). To characterize the column space, apply $\mathbf{A}$ to a generic vector $\mathbf{x} \in \mathbb{R}^n$:

$$\mathbf{A}\mathbf{x} = \Big(\sum_{j=1}^{r} \sigma_j \mathbf{u}_j \mathbf{v}_j^T \Big) \mathbf{x} = \sum_{j=1}^{r} \left(\sigma_j \mathbf{u}_j \mathbf{v}_j^T \mathbf{x} \right) = \sum_{j=1}^{r} \left(\sigma_j \mathbf{v}_j^T \mathbf{x} \right) \mathbf{u}_j, \tag{6.9}$$

where in the last step we have switched the order of the scalar $\mathbf{v}_j^T \mathbf{x}$ and the vector $\mathbf{u}_j$. We see that $\mathbf{A}\mathbf{x}$ is a weighted sum of the vectors $\mathbf{u}_1, \ldots, \mathbf{u}_r$. Since this must hold for all $\mathbf{x} \in \mathbb{R}^n$, we conclude that

$$\mathcal{R}(\mathbf{A}) \subseteq \operatorname{span}\{\mathbf{u}_1, \ldots, \mathbf{u}_r\}.$$

Can we conclude the converse? We know that $\mathcal{R}(\mathbf{A})$ is a subspace, so if we can show that each of the vectors $\mathbf{u}_1, \ldots, \mathbf{u}_r$ is in $\mathcal{R}(\mathbf{A})$, then we will know that

$$\operatorname{span}\{\mathbf{u}_1, \ldots, \mathbf{u}_r\} \subseteq \mathcal{R}(\mathbf{A}). \tag{6.10}$$

To show that $\mathbf{u}_k \in \mathcal{R}(\mathbf{A})$, we must find some $\mathbf{x}$ such that $\mathbf{A}\mathbf{x} = \mathbf{u}_k$. Inspect equation (6.9). We can make $\mathbf{A}\mathbf{x} = \mathbf{u}_k$ if all the coefficients $\sigma_j \mathbf{v}_j^T \mathbf{x}$ are zero when $j \neq k$, and $\sigma_k \mathbf{v}_k^T \mathbf{x} = 1$. Can you see how to use orthogonality of the right singular vectors $\mathbf{v}_1, \ldots, \mathbf{v}_r$ to achieve this? Setting

$$\mathbf{x} = \frac{1}{\sigma_k} \mathbf{v}_k,$$

we have $\mathbf{A}\mathbf{x} = \mathbf{u}_k$. Thus $\mathbf{u}_k \in \mathcal{R}(\mathbf{A})$, and we can conclude that (6.10) holds. Since $\mathcal{R}(\mathbf{A})$ and $\operatorname{span}\{\mathbf{u}_1, \ldots, \mathbf{u}_r\}$ contain one another, we conclude that

$$\mathcal{R}(\mathbf{A}) = \operatorname{span}\{\mathbf{u}_1, \ldots, \mathbf{u}_r\}.$$

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

$$\mathbf{A}^T = \Big(\sum_{j=1}^{r} \sigma_j \mathbf{u}_j \mathbf{v}_j^T \Big)^T = \sum_{j=1}^{r} \Big(\sigma_j \mathbf{u}_j \mathbf{v}_j^T \Big)^T = \sum_{j=1}^{r} \sigma_j \mathbf{v}_j \mathbf{u}_j^T.$$

Adapting the argument we have just made leads to

$$\mathcal{R}(\mathbf{A}^T) = \operatorname{span}\{\mathbf{v}_1, \ldots, \mathbf{v}_r\}.$$

Equation (6.9) for $\mathbf{A}\mathbf{x}$ is also the key that unlocks the null space $\mathcal{N}(\mathbf{A})$. For what $\mathbf{x} \in \mathbb{R}^n$ does $\mathbf{A}\mathbf{x} = \mathbf{0}$? Let us consider

$$\begin{aligned} \|\mathbf{A}\mathbf{x}\|^2 = (\mathbf{A}\mathbf{x})^T (\mathbf{A}\mathbf{x}) &= \Big(\sum_{j=1}^{r} \left(\sigma_j \mathbf{v}_j^T \mathbf{x} \right) \mathbf{u}_j \Big)^T \Big(\sum_{k=1}^{r} \left(\sigma_k \mathbf{v}_k^T \mathbf{x} \right) \mathbf{u}_k \Big) \\ &= \Big(\sum_{j=1}^{r} \left(\sigma_j \mathbf{x}^T \mathbf{v}_j \right) \mathbf{u}_j^T \Big) \Big(\sum_{k=1}^{r} \left(\sigma_k \mathbf{v}_k^T \mathbf{x} \right) \mathbf{u}_k \Big) \\ &= \sum_{j=1}^{r} \sum_{k=1}^{r} \Big(\left(\sigma_j \mathbf{x}^T \mathbf{v}_j \right) \left(\sigma_k \mathbf{v}_k^T \mathbf{x} \right) \mathbf{u}_j^T \mathbf{u}_k \Big). \end{aligned}$$

Notice that $\mathbf{v}_j^T \mathbf{x} = (\mathbf{v}_j^T \mathbf{x})^T = \mathbf{x}^T \mathbf{v}_j$ because $\mathbf{v}_j^T \mathbf{x}$ is a real-valued scalar number.

Since the left singular vectors are orthogonal, $\mathbf{u}_j^T \mathbf{u}_k = 0$ for $j \neq k$, this double-sum collapses: only the terms with $j = k$ make a nontrivial contribution:

$$\|\mathbf{A}\mathbf{x}\|^2 = \sum_{j=1}^{r} \left(\sigma_j \mathbf{x}^T \mathbf{v}_j\right)\left(\sigma_j \mathbf{v}_j^T \mathbf{x}\right)\mathbf{u}_j^T \mathbf{u}_j = \sum_{j=1}^{r} \sigma_j^2 |\mathbf{v}_j^T \mathbf{x}|^2, \tag{6.11}$$

since $\mathbf{u}_j^T \mathbf{u}_j = 1$ and $(\mathbf{x}^T \mathbf{v}_j)(\mathbf{v}_j^T \mathbf{x}) = |\mathbf{v}_j^T \mathbf{x}|^2$.

Since $\sigma_j > 0$, the right-hand side of (6.11) is the sum of nonnegative numbers. To have $\|\mathbf{A}\mathbf{x}\| = 0$, all the coefficients in this sum must be zero. The only way for that to happen is for

$$\mathbf{v}_j^T \mathbf{x} = 0, \qquad j = 1, \ldots, r,$$

i.e., $\mathbf{A}\mathbf{x} = \mathbf{0}$ if and only if $\mathbf{x}$ is orthogonal to $\mathbf{v}_1, \ldots, \mathbf{v}_r$. We already have a characterization of such vectors from the singular value decomposition:

$$\mathbf{x} \in \operatorname{span}\{\mathbf{v}_{r+1}, \ldots, \mathbf{v}_n\}.$$

Thus we conclude

$$\mathcal{N}(\mathbf{A}) = \operatorname{span}\{\mathbf{v}_{r+1}, \ldots, \mathbf{v}_n\}.$$

If $r = n$, then this span is vacuous, and we just have $\mathcal{N}(\mathbf{A}) = \mathbf{0}$.

To compute $\mathcal{N}(\mathbf{A}^T)$, we can repeat the same argument based on $\|\mathbf{A}^T\mathbf{y}\|^2$ to obtain

$$\mathcal{N}(\mathbf{A}^T) = \operatorname{span}\{\mathbf{u}_{r+1}, \ldots, \mathbf{u}_m\}.$$

Putting these results together, we arrive at a beautiful elaboration of the Fundamental Theorem of Linear Algebra[1].

[1] Gilbert Strang. The Fundamental Theorem of Linear Algebra. *Amer. Math. Monthly*, 100:848–855, 1993

Theorem 14 (Fundamental Theorem of Linear Algebra, SVD Version) *Suppose* $\mathbf{A} \in \mathbb{R}^{m\times n}$ *has* $\operatorname{rank}(\mathbf{A}) = r$*, with left singular vectors* $\{\mathbf{u}_1, \ldots, \mathbf{u}_m\}$ *and right singular vectors* $\{\mathbf{v}_1, \ldots, \mathbf{v}_n\}$*. Then*

$$\begin{aligned}
\mathcal{R}(\mathbf{A}) &= \operatorname{span}\{\mathbf{u}_1, \ldots, \mathbf{u}_r\} \\
\mathcal{N}(\mathbf{A}^T) &= \operatorname{span}\{\mathbf{u}_{r+1}, \ldots, \mathbf{u}_m\} \\
\mathcal{R}(\mathbf{A}^T) &= \operatorname{span}\{\mathbf{v}_1, \ldots, \mathbf{v}_r\} \\
\mathcal{N}(\mathbf{A}) &= \operatorname{span}\{\mathbf{v}_{r+1}, \ldots, \mathbf{v}_n\},
\end{aligned}$$

which implies

$$\begin{aligned}
\mathcal{R}(\mathbf{A}) \oplus \mathcal{N}(\mathbf{A}^T) &= \operatorname{span}\{\mathbf{u}_1, \ldots, \mathbf{u}_m\} = \mathbb{R}^m \\
\mathcal{R}(\mathbf{A}^T) \oplus \mathcal{N}(\mathbf{A}) &= \operatorname{span}\{\mathbf{v}_1, \ldots, \mathbf{v}_n\} \;\; = \mathbb{R}^n,
\end{aligned}$$

and

$$\mathcal{R}(\mathbf{A}) \perp \mathcal{N}(\mathbf{A}^T), \qquad \mathcal{R}(\mathbf{A}^T) \perp \mathcal{N}(\mathbf{A}).$$

6.10 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 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 $\mathbf{A} \in \mathbb{R}^{m\times n}$ by the maximum amount it can stretch a vector, $\mathbf{x} \in \mathbb{R}^n$. 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 (6.11); however, we will take a slightly different approach. First, suppose that $\mathbf{Q}$ is some matrix with orthonormal columns, so that $\mathbf{Q}^T\mathbf{Q} = \mathbf{I}$. Then

$$\|\mathbf{Q}\mathbf{x}\|^2 = (\mathbf{Q}\mathbf{x})^T(\mathbf{Q}\mathbf{x}) = \mathbf{x}^T\mathbf{Q}^T\mathbf{Q}\mathbf{x} = \mathbf{x}^T\mathbf{x} = \|\mathbf{x}\|^2,$$

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

$$\|\mathbf{A}\mathbf{x}\| = \|\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T\mathbf{x}\| = \|\mathbf{\Sigma}\mathbf{V}^T\mathbf{x}\|,$$

where we have used the orthonormality of the columns of $\mathbf{U}$. Now define a new variable $\mathbf{y} = \mathbf{V}^T\mathbf{x}$ (which means $\mathbf{V}\mathbf{y} = \mathbf{x}$), and notice that $\|\mathbf{x}\| = \|\mathbf{V}^T\mathbf{x}\| = \|\mathbf{y}\|$, since $\mathbf{V}$ is a square matrix with orthonormal columns (and hence orthonormal rows). Now we can compute the matrix norm:

The fact that $\mathbf{V}$ is square and has orthonormal columns implies that both $\mathbf{V}^T\mathbf{V} = \mathbf{I}$ and $\mathbf{V}\mathbf{V}^T = \mathbf{I}$. This means that $\|\mathbf{V}^T\mathbf{x}\|^2 = \mathbf{x}^T\mathbf{V}\mathbf{V}^T\mathbf{x} = \mathbf{x}^T\mathbf{x} = \|\mathbf{x}\|^2$.

$$\|\mathbf{A}\| = \max_{\|\mathbf{x}\|=1} \|\mathbf{A}\mathbf{x}\| = \max_{\|\mathbf{x}\|=1} \|\mathbf{\Sigma}\mathbf{V}^T\mathbf{x}\| = \max_{\|\mathbf{V}\mathbf{y}\|=1} \|\mathbf{\Sigma}\mathbf{y}\| = \max_{\|\mathbf{y}\|=1} \|\mathbf{\Sigma}\mathbf{y}\|$$

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 $\sigma_1 \geq \cdots \geq \sigma_r$,

Alternatively, you could compute $\|\mathbf{\Sigma}\|$ by maximizing $f(\mathbf{y}) = \|\mathbf{\Sigma}\mathbf{y}\|$ subject to $\|\mathbf{y}\| = 1$ using the Lagrange multiplier technique from vector calculus.

$$\begin{aligned}\|\mathbf{\Sigma}\mathbf{y}\|^2 &= \sigma_1^2|y_1|^2 + \cdots + \sigma_r^2|y_r|^2 \\ &\leq \sigma_1^2\Big(|y_1|^2 + \cdots + |y_r|^2\Big) \leq \sigma_1^2\|\mathbf{y}\|^2 = \sigma_1^2,\end{aligned}$$

resulting in the upper bound

$$\|\mathbf{\Sigma}\| = \max_{\|\mathbf{y}\|=1} \|\mathbf{\Sigma}\mathbf{y}\| \leq \sigma_1. \tag{6.12}$$

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

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

Since $|\mathbf{\Sigma y}\|$ can be no larger than σ_1 for any $\mathbf{y}$, and since $\|\mathbf{\Sigma y}\| = \sigma_1$ for at least one choice of $\mathbf{y}$, we conclude

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

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

$$\|\mathbf{A}\| = \sigma_1.$$

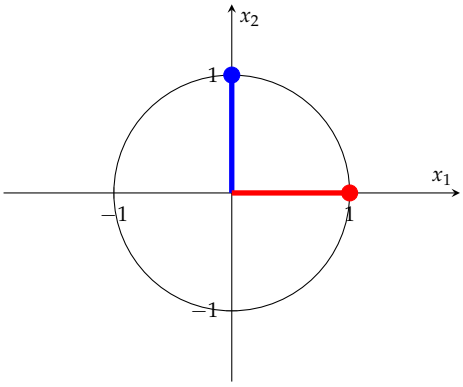
Consider the matrix

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

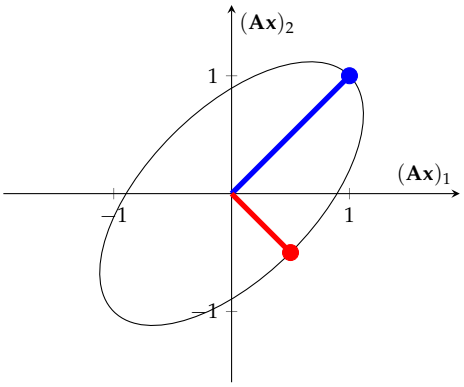
We see from this SVD that $\|\mathbf{A}\| = \sigma_1 = \sqrt{2}$. For this example the vector $\mathbf{Ax}$ has the form

$$\begin{aligned} \mathbf{Ax} &= \sigma_1 (\mathbf{v}_1^T \mathbf{x}) \mathbf{u}_1 + \sigma_2 (\mathbf{v}_2^T \mathbf{x}) \mathbf{u}_2 \\ &= \sqrt{2} x_2 \mathbf{u}_1 + \frac{\sqrt{2}}{2} x_1 \mathbf{u}_2, \end{aligned}$$

so $\mathbf{Ax}$ is a blend of some expansion in the $\mathbf{u}_1$ direction and some contraction in the $\mathbf{u}_2$ direction. We maximize the size of $\mathbf{Ax}$ by picking an $\mathbf{x}$ for which $\mathbf{Ax}$ is maximally rich in $\mathbf{u}_1$, i.e., $\mathbf{x} = \mathbf{v}_1$.



Every unit vector $\mathbf{x}$ in $\mathbb{R}^2$ is a point where $\|\mathbf{x}\|^2 = x_1^2 + x_2^2 = 1$, so the set of all such vectors traces out the *unit circle* shown in black in the plot above. We highlight two distinguished vectors: $\mathbf{x} = \mathbf{v}_1$ (blue) and $\mathbf{x} = \mathbf{v}_2$ (red).



The plot above shows $\mathbf{Ax}$ for all unit vectors $\mathbf{x}$, which traces out an ellipse in $\mathbb{R}^2$. The vector $\mathbf{x} = \mathbf{v}_1$ is mapped to $\mathbf{Ax} = \sigma_1 \mathbf{u}_1$ (blue), and this is the most $\mathbf{A}$ stretches any unit vector; $\mathbf{x} = \mathbf{v}_2$ is mapped to $\mathbf{Ax} = \sigma_2 \mathbf{u}_2$ (red), which gives the smallest value of $\|\mathbf{Ax}\|$. (Plots like this can be traced out with MATLAB's `eigshow` command.)

6.11 *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

$$\mathbf{A} = \sum_{j=1}^{r} \sigma_j \mathbf{u}_j \mathbf{v}_j^T$$

writes the rank-r matrix $\mathbf{A}$ as the sum of the r rank-1 matrices

$$\sigma_j \mathbf{u}_j \mathbf{v}_j^T.$$

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

$$\sum_{j=1}^{k} \sigma_j \mathbf{u}_j \mathbf{v}_j^T$$

will give a good approximation to $\mathbf{A}$ 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 $\mathbf{A}$ models some low-rank phenomenon, but some noise (such as random sampling errors, when the entries of $\mathbf{A}$ are measured from some physical process) causes $\mathbf{A}$ to have much larger rank. If the noise is small relative to the "true" data in $\mathbf{A}$, we expect $\mathbf{A}$ to have a number of very small singular values that we might wish to neglect as we work with $\mathbf{A}$. We will see examples of this kind of behavior in the next chapter.

For square diagonalizable matrices,

$$\mathbf{A} = \mathbf{W}\mathbf{\Lambda}\mathbf{W}^{-1},$$

the eigenvalue decomposition can also lead to an expression for $\mathbf{A}$ as the sum of rank-1 matrices:

$$\mathbf{A} = \sum_{j=1}^{n} \lambda_j \mathbf{w}_j \widehat{\mathbf{w}}_j^T,$$

where $\widehat{\mathbf{w}}_j^T$ denotes the jth row of $\mathbf{W}$, and we have used the conjugate-transpose because the eigenvector might have complex entries, even when $\mathbf{A}$ only has real entries.

Note that if $\mathbf{A}$ has real entries, then the SVD will only have real entries. This is not generally the case for the eigenvalue decomposition when $\mathbf{A}$ is a nonsymmetric matrix.

Three key distinctions make the singular value decomposition a better tool for developing low-rank approximations to $\mathbf{A}$.

1. The SVD holds for all matrices, while the eigenvalue decomposition only holds for square matrices that are diagonalizable.

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 $\sigma_j \mathbf{u}_j \mathbf{v}_j^T$ contribute to $\mathbf{A}$. In contrast, the eigenvalues will generally be complex numbers, and thus do not have the same natural order; it is harder to understand the significance of each rank-1 matrix $\lambda_j \mathbf{w}_j \widehat{\mathbf{w}}_j^T$.

3. The eigenvectors are not generally orthogonal, and this can skew the rank-1 matrices $\lambda_j \mathbf{w}_j \widehat{\mathbf{w}}_j^T$ away from giving good approximations to $\mathbf{A}$. In particular, we can find that $\|\mathbf{w}_j \widehat{\mathbf{w}}_j^T\| \gg 1$, whereas the matrices $\mathbf{u}_j \mathbf{v}_j^T$ from the SVD always satisfy $\|\mathbf{u}_j \mathbf{v}_j^T\| = 1$.

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

$$\mathbf{A} = \begin{bmatrix} 2 & 100 \\ 0 & 1 \end{bmatrix}$$

with eigenvalues $\lambda_1 = 2$ and $\lambda_2 = 1$ and eigenvalue decomposition

$$\begin{aligned}\mathbf{A} = \mathbf{W}\mathbf{\Lambda}\mathbf{W}^{-1} &= \begin{bmatrix} 1 & 1 \\ 0 & -1/100 \end{bmatrix} \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 100 \\ 0 & -100 \end{bmatrix} \\ &= \lambda_1 \mathbf{w}_1 \widehat{\mathbf{w}}_1^T + \lambda_2 \mathbf{w}_2 \widehat{\mathbf{w}}_2^T \\ &= 2 \begin{bmatrix} 1 \\ 0 \end{bmatrix} \begin{bmatrix} 1 & 100 \end{bmatrix} + 1 \begin{bmatrix} 1 \\ -1/100 \end{bmatrix} \begin{bmatrix} 0 & -100 \end{bmatrix} \\ &= 2 \begin{bmatrix} 1 & 100 \\ 0 & 0 \end{bmatrix} + 1 \begin{bmatrix} 0 & -100 \\ 0 & 1 \end{bmatrix}.\end{aligned}$$

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

$$\lambda_1 \mathbf{w}_1 \widehat{\mathbf{w}}_1^T = \begin{bmatrix} 2 & 200 \\ 0 & 0 \end{bmatrix}, \qquad \lambda_2 \mathbf{w}_2 \widehat{\mathbf{w}}_2^T = \begin{bmatrix} 0 & -100 \\ 0 & 1 \end{bmatrix}.$$

Neither matrix individually gives a good approximation to $\mathbf{A}$*:*

$$\mathbf{A} - \lambda_1 \mathbf{w}_1 \widehat{\mathbf{w}}_1^T = \begin{bmatrix} 0 & -100 \\ 0 & 1 \end{bmatrix}, \qquad \mathbf{A} - \lambda_2 \mathbf{w}_2 \widehat{\mathbf{w}}_2^T = \begin{bmatrix} 2 & 200 \\ 0 & 0 \end{bmatrix}.$$

Both rank-1 "approximations" to $\mathbf{A}$ leave large errors!

Contrast this situation with the rank-1 approximation $\sigma_1 \mathbf{u}_1 \mathbf{v}_1^T$ given by the SVD for this $\mathbf{A}$. To five decimal digits, we have

$$\begin{aligned}\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T &= \begin{bmatrix} 0.99995 & -0.01000 \\ 0.01000 & 0.99995 \end{bmatrix} \begin{bmatrix} 100.025 & 0 \\ 0 & 0.020 \end{bmatrix} \begin{bmatrix} 0.01999 & 0.99980 \\ -0.99980 & 0.01999 \end{bmatrix} \\ &= \sigma_1 \mathbf{u}_1 \mathbf{v}_1^T + \ +\sigma_2 \mathbf{u}_2 \mathbf{v}_2^T \\ &= 100.025 \begin{bmatrix} 0.99995 \\ 0.01000 \end{bmatrix} \begin{bmatrix} 0.01999 & 0.99980 \end{bmatrix} + 0.020 \begin{bmatrix} -0.01000 \\ 0.99995 \end{bmatrix} \begin{bmatrix} -0.99980 & 0.01999 \end{bmatrix} \\ &= 100.025 \begin{bmatrix} 0.01999 & 0.99975 \\ 0.00020 & 00.00999 \end{bmatrix} + 0.020 \begin{bmatrix} 0.00999 & -0.00020 \\ -.99975 & 0.01999 \end{bmatrix}.\end{aligned}$$

Like the eigendecomposition, the SVD breaks $\mathbf{A}$ into two rank-1 pieces:

$$\sigma_1 \mathbf{u}_1 \mathbf{v}_1^T = \begin{bmatrix} 1.99980 & 100.00000 \\ 0.01999 & 0.99960 \end{bmatrix}, \qquad \sigma_2 \mathbf{u}_2 \mathbf{v}_2^T = \begin{bmatrix} 0.00020 & 0.00000 \\ -0.01999 & 0.00040 \end{bmatrix}.$$

The first of these, the dominant term in the SVD, gives an *excellent* approximation to $\mathbf{A}$:

$$\mathbf{A} - \sigma_1 \mathbf{u}_1 \mathbf{v}_1^T = \begin{bmatrix} 0.00020 & 0.00000 \\ -0.01999 & 0.00040 \end{bmatrix}.$$

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 23 *Let $\mathbf{A} = \sum_{j=1}^{r} \sigma_j \mathbf{u}_j \mathbf{v}_j^T$ 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*

$$\mathbf{A}_k = \sum_{j=1}^{k} \sigma_j \mathbf{u}_j \mathbf{v}_j^T.$$

Theorem 15 (Schmidt–Mirsky–Eckart–Young) *Let $\mathbf{A} \in \mathbb{R}^{m \times n}$. Then for all $k \leq \text{rank}(\mathbf{A})$, the truncated singular value decomposition*

$$\mathbf{A}_k = \sum_{j=1}^{k} \sigma_j \mathbf{u}_j \mathbf{v}_j^T$$

is a best rank-k approximation to $\mathbf{A}$, in the sense that

$$\|\mathbf{A} - \mathbf{A}_k\| = \min_{\text{rank}(\mathbf{X}) \leq k} \|\mathbf{A} - \mathbf{X}\| = \sigma_{k+1}.$$

It is easy to see that this $\mathbf{A}_k$ gives the approximation error σ_{k+1}, since

$$\mathbf{A} - \mathbf{A}_k = \sum_{j=1}^{r} \sigma_j \mathbf{u}_j \mathbf{v}_j^T - \sum_{j=1}^{k} \sigma_j \mathbf{u}_j \mathbf{v}_j^T = \sum_{j=k+1}^{r} \sigma_j \mathbf{u}_j \mathbf{v}_j^T,$$

and this last expression is an SVD for the error in the approximation $\mathbf{A} - \mathbf{A}_k$. As described in Section 6.10, the norm of a matrix equals its largest singular value, so

$$\|\mathbf{A} - \mathbf{A}_k\| = \left\| \sum_{j=k+1}^{r} \sigma_j \mathbf{u}_j \mathbf{v}_k^T \right\| = \sigma_{k+1}.$$

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

6.11.1 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.)

Let $\mathbf{X} \in \mathbb{R}^{m \times n}$ be any rank-k matrix. The Fundamental Theorem of Linear Algebra gives $\mathbb{R}^n = \mathcal{R}(\mathbf{X}^T) \oplus \mathcal{N}(\mathbf{X})$. Since $\text{rank}(\mathbf{X}^T) = \text{rank}(\mathbf{X}) = k$, notice that $\dim(\mathcal{N}(\mathbf{X})) = n - k$. From the singular value decomposition of $\mathbf{A}$ extract $\mathbf{v}_1, \ldots, \mathbf{v}_{k+1}$, a basis for some $k+1$ dimensional subspace of $\mathbb{R}^n$. Since $\mathcal{N}(\mathbf{X}) \subseteq \mathbb{R}^n$ has dimension $n - k$, it must be that the intersection

$$\mathcal{N}(\mathbf{X}) \cap \text{span}\{\mathbf{v}_1, \ldots, \mathbf{v}_{k+1}\}$$

has dimension at least one. (Otherwise, $\mathcal{N}(\mathbf{X}) \oplus \text{span}\{\mathbf{v}_1, \ldots, \mathbf{v}_{k+1}\}$ would be an $n+1$ dimensional subspace of $\mathbb{R}^n$: impossible!) Let $\mathbf{z}$ be some unit vector in that intersection: $\|\mathbf{z}\| = 1$ and

$$\mathbf{z} \in \mathcal{N}(\mathbf{X}) \cap \text{span}\{\mathbf{v}_1, \ldots, \mathbf{v}_{k+1}\}.$$

Expand $\mathbf{z} = \gamma_1 \mathbf{v}_1 + \cdots + \gamma_{k+1} \mathbf{v}_{k+1}$, so that $\|\mathbf{z}\| = 1$ implies

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

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

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

and then

$$\|\mathbf{A}\mathbf{z}\| = \left\| \sum_{j=1}^{k+1} \sigma_j \mathbf{u}_j \mathbf{v}_j^T \mathbf{z} \right\| = \left\| \sum_{j=1}^{k+1} \sigma_j \gamma_j \mathbf{u}_j \right\|.$$

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

$$\left\| \sum_{j=1}^{k+1} \sigma_j \gamma_j \mathbf{u}_j \right\|_2 \geq \sigma_{k+1} \left\| \sum_{j=1}^{k+1} \gamma_j \mathbf{u}_j \right\|_2.$$

But notice that

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

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

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

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

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:

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

The `double` command converts the entries of the image into floating point numbers. If your file contains a grayscale image, `A` will typically contain the $m \times n$ matrix containing the gray colors of your image; however, you might need to extract this level from an $m \times n \times 3$ *tensor*:

To conserve memory, MATLAB's default is to save the entries of an image as integers, but MATLAB's linear algebra routines like `svd` only work with floating point matrices.

```
A= A(:,:,1);
```

If you have a color image, then `A` will be an $m \times n \times 3$ tensor, with each slice describing the intensity of one of the primary colors. Extract these red, green, and blue color matrices in an extra step:

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

You might be wondering: Could we just take the SVD of the $m \times n \times 3$ tensor directly? This fascinating topic of low-rank tensor approximation has received much research attention over the past twenty years. For details, see: Tamara G. Kolda and Brett W. Bader, "Tensor decompositions and applications," *SIAM Review* 51 (2009) 455–500.

Finally, to visualize an image in MATLAB, use

```
imagesc(A)
```

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 $\mathbf{A}$ 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 $\mathbf{A}$ can be represented well by a rank-k matrix, then one can approximate $\mathbf{A}$ by storing only the leading k singular values and vectors. To build this approximation

$$\mathbf{A}_k = \sum_{j=1}^{k} \sigma_j \mathbf{u}_j \mathbf{v}_j^T,$$

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 $\mathbf{A}$.

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 algebra algorithms we have studied this semester, gathered in Gatlinburg, Tennessee, for an important early conference in the field. The image

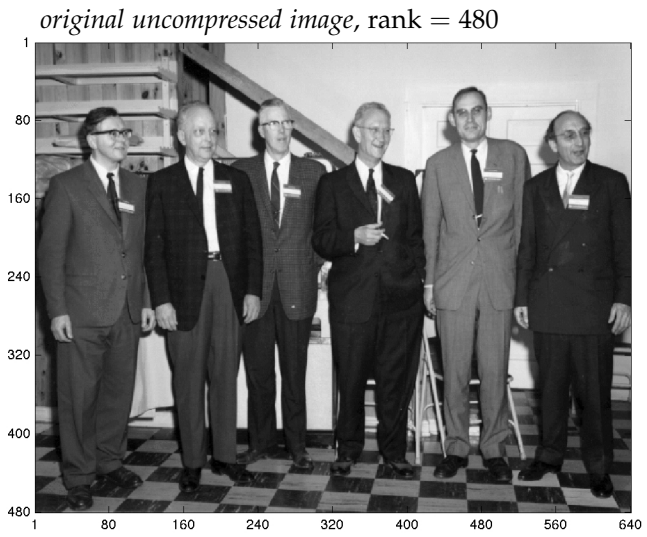


Figure 6.1: A sample image: the founders of numerical linear algebra at an early Gatlinburg Symposium. From left to right: Jim Wilkinson, Wallace Givens, George Forsythe, Alston Householder, Peter Henrici, and Friedrich Bauer.

is of size 480×640, so $\text{rank}(\mathbf{A}) \leq 480$. We shall compress this image with truncated singular value decompositions. Figures 6.3 and 6.4 show compressions of $\mathbf{A}$ for dimensions ranging from $k = 200$ down to $k = 1$. For $k = 200$ and 100, the compression $\mathbf{A}_k$ provides an excellent proxy for the full image $\mathbf{A}$. 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 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 $\mathbf{A}$, shown in Figure 6.2. The first sin-

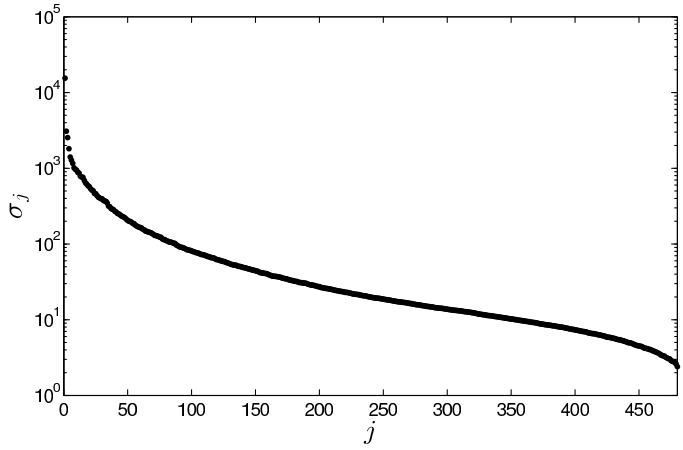
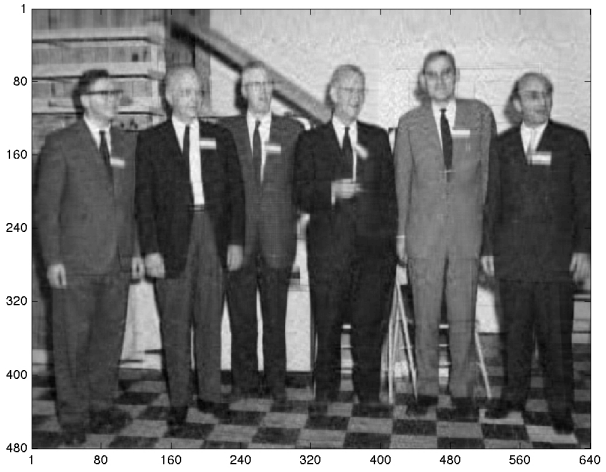


Figure 6.2: Singular values of the 480×640 Gatlinburg image matrix. The first few singular values are much larger than the rest, suggesting the potential for accurate low-rank approximation (compression).

truncated SVD, rank $k = 200$

truncated SVD, rank $k = 100$

truncated SVD, rank $k = 50$



truncated SVD, rank $k = 25$

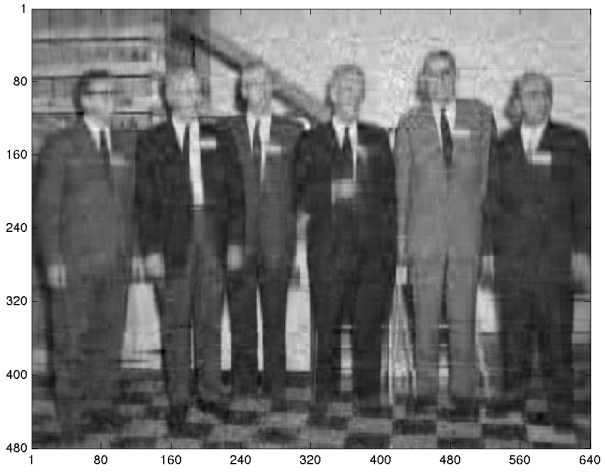


Figure 6.3: Compressions of the Gatlinburg image in Figure 6.1 using truncated SVDs $\mathbf{A}_k = \sum_{j=1}^{k} \sigma_j \mathbf{u}_j \mathbf{v}_j^T$. Each of these images can be stored with less memory than the original full image. The rank-25 image could be be useful as a "thumbnail" sketch of the image (e.g., an icon on a computer desktop).

gular value σ_1 is a 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.

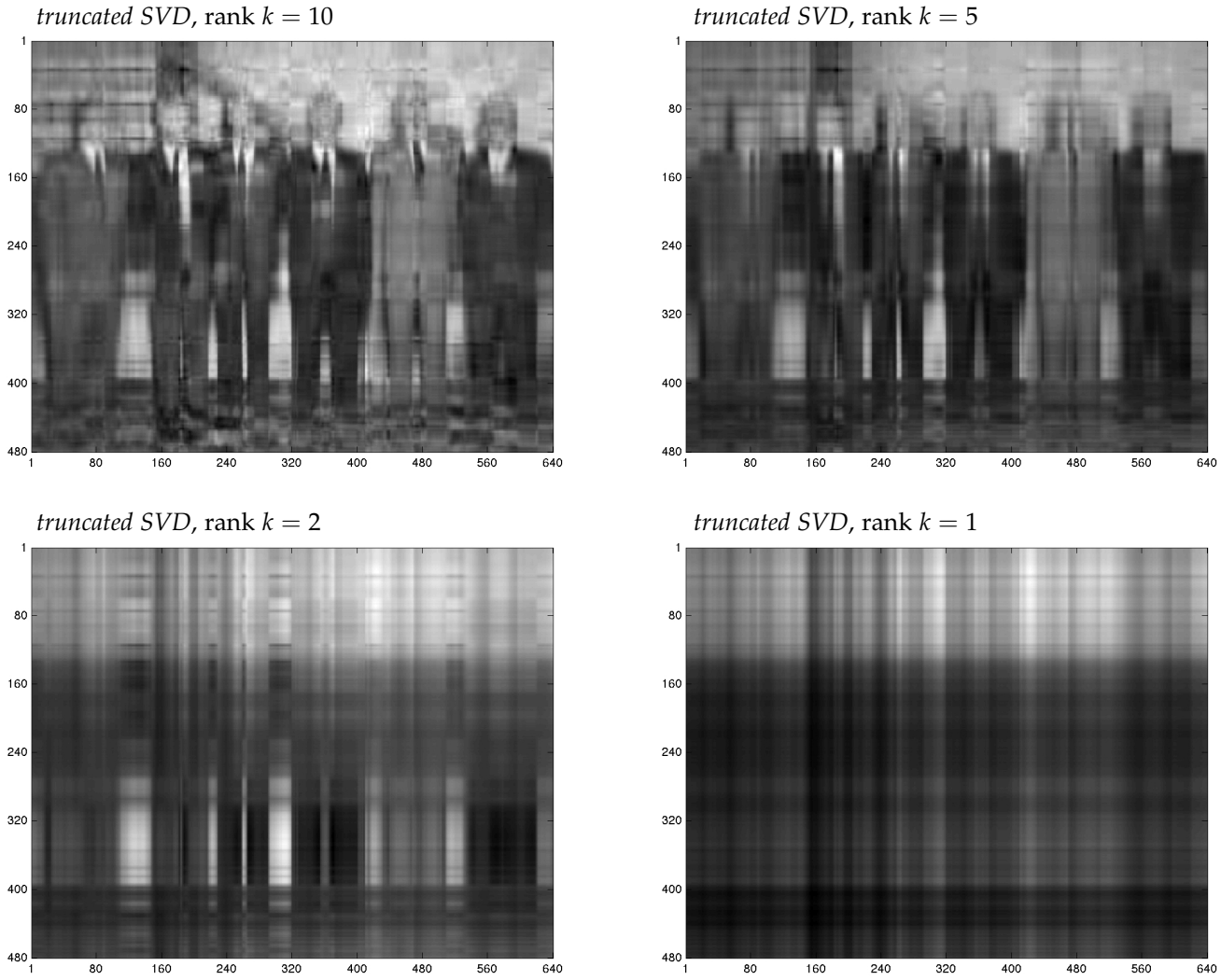


Figure 6.4: Continuation of Figure 6.4, showing compressions of rank 10, 5, 2, and 1. Note the striping characteristic of low-rank structure.

To investigate this low-rank approximation a little more deeply, let us introduce another image, a carved grotesque, shown in Figure 6.5. This grayscale image comprises 644×500 pixels, suggesting that $\text{rank}(\mathbf{A}) = 500$. Figure 6.6 shows that singular values decay much like those for the Gatlinburg matrix (Figure 6.2); indeed, the grotesque's first singular value is at least ten times larger than all the others, with $\sigma_1 \approx 8.84 \times 10^4$ while $\sigma_2 \approx 7.91 \times 10^3$.

Based on these singular values, we expect a strong low-rank approximation. Figure 6.7 shows rank-k truncated SVD approximations for eight values of k. Indeed, given the dominant size of σ_1, we see the major structure of the frame and border evident even in the $k = 1$ compression.

To emphasize how the individual components $\sigma_j \mathbf{u}_j \mathbf{v}^T$ contribute to

Figure 6.5: A grotesque carved in a door of the 16th century Church of Santa Croce, Riva San Vitale, Switzerland.

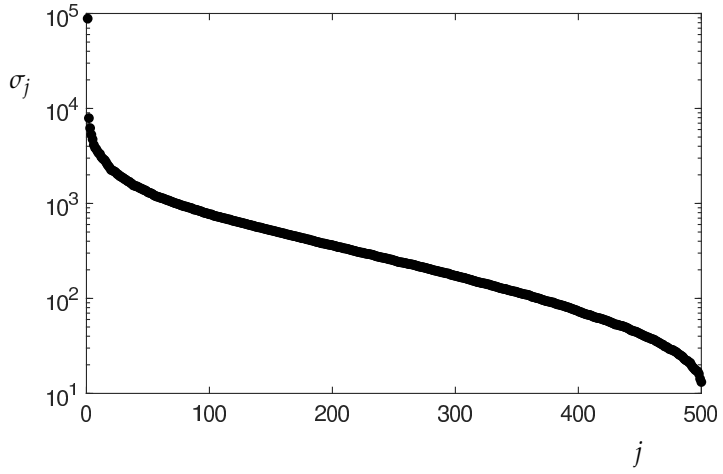


Figure 6.6: Singular values of the 644×500 Santa Croce grotesque. The first singular value is ten times or more larger than the others.

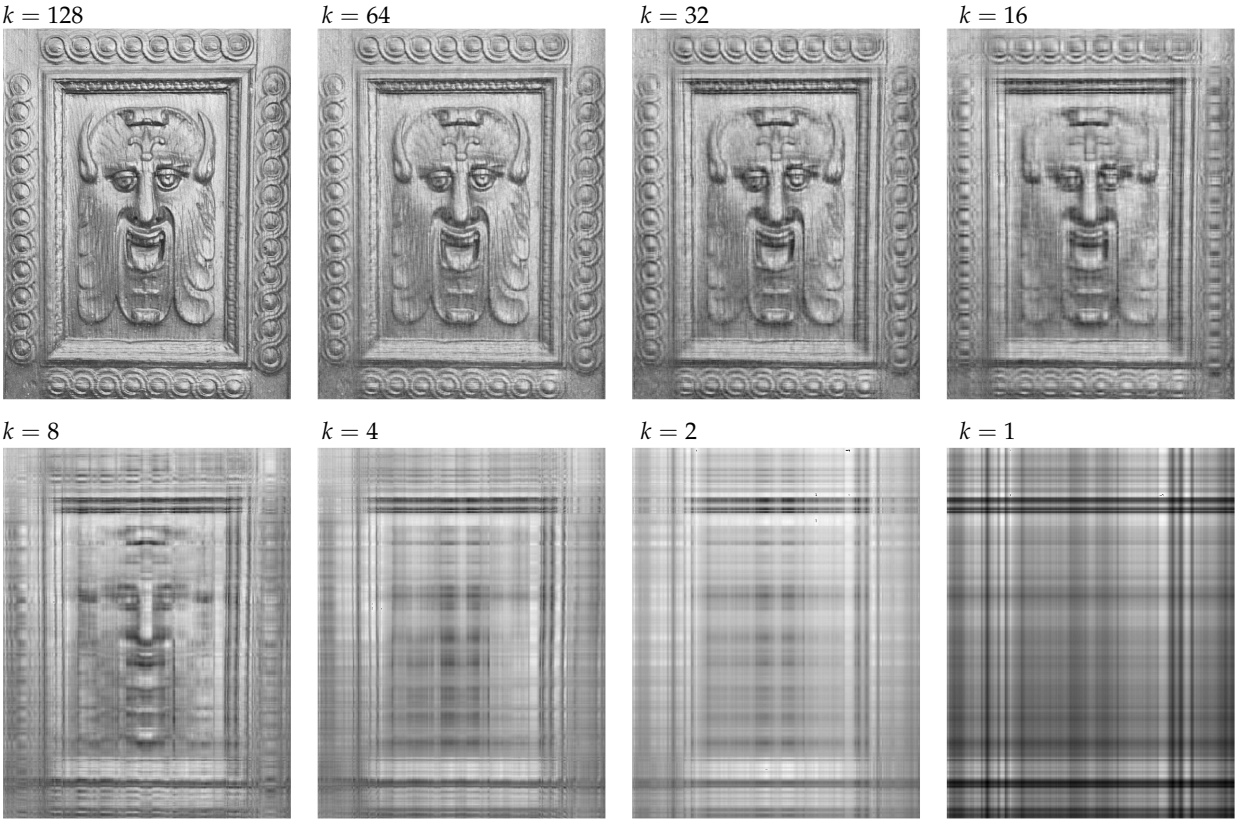


Figure 6.7: Rank-k truncated SVD compressed versions of the Santa Croce grotesque. (Close examination of the original image shows numerous worm holes, especially in the lower left of the panel. Notice how these are reduced when $k = 128$ and essentially disappear when $k = 64$.)

the sum

$$\mathbf{A} = \sum_{j=1}^{r} \sigma_j \mathbf{u}_j \mathbf{v}^T, \tag{6.13}$$

we shall take a closer look at the image in Figure 6.5. To make this point as clearly as possible, we introduce a new color map that shows

positive values in blue and negative values in red, as illustrated in Figure 6.8. (The image $\mathbf{A}$ has integer entries between $[0, 255]$, but the truncated SVDs $\mathbf{A}_k$ need not have integer entries, and the individual terms $\sigma_j \mathbf{u}_j \mathbf{v}_j^T$ can have negative entries.)

Figure 6.9 shows how this image is assembled from the individual terms in the dyadic form of the singular value decomposition (6.13). Since $\sigma_1 \approx 8.84 \times 10^4$ is so much larger than $\sigma_2 \approx 7.91 \times 10^3$, the first term dominates (hence the dark blue color): the interior frame around the face is already evident. The subsequent matrices $\sigma_j \mathbf{u}_j \mathbf{v}_j^T$ for $j \geq 2$ add more modest corrections that fill in details of the image. Some of these effects can be readily picked out: for example, $\sigma_3 \mathbf{u}_3 \mathbf{v}_3^T$ adjusts for the row of carving at the top of the image; $\sigma_5 \mathbf{u}_5 \mathbf{v}_5^T$ fills in the grotesque's nose. Since the singular values σ_j are decreasing as j grows, these terms make smaller and smaller contributions.

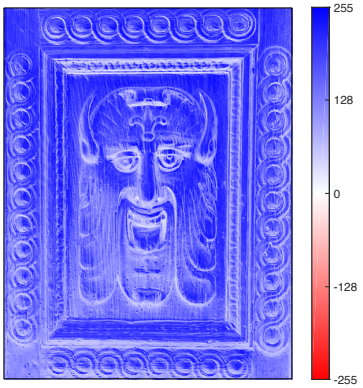


Figure 6.8: The Santa Croce grotesque with an extended color map to highlight negative values (red).

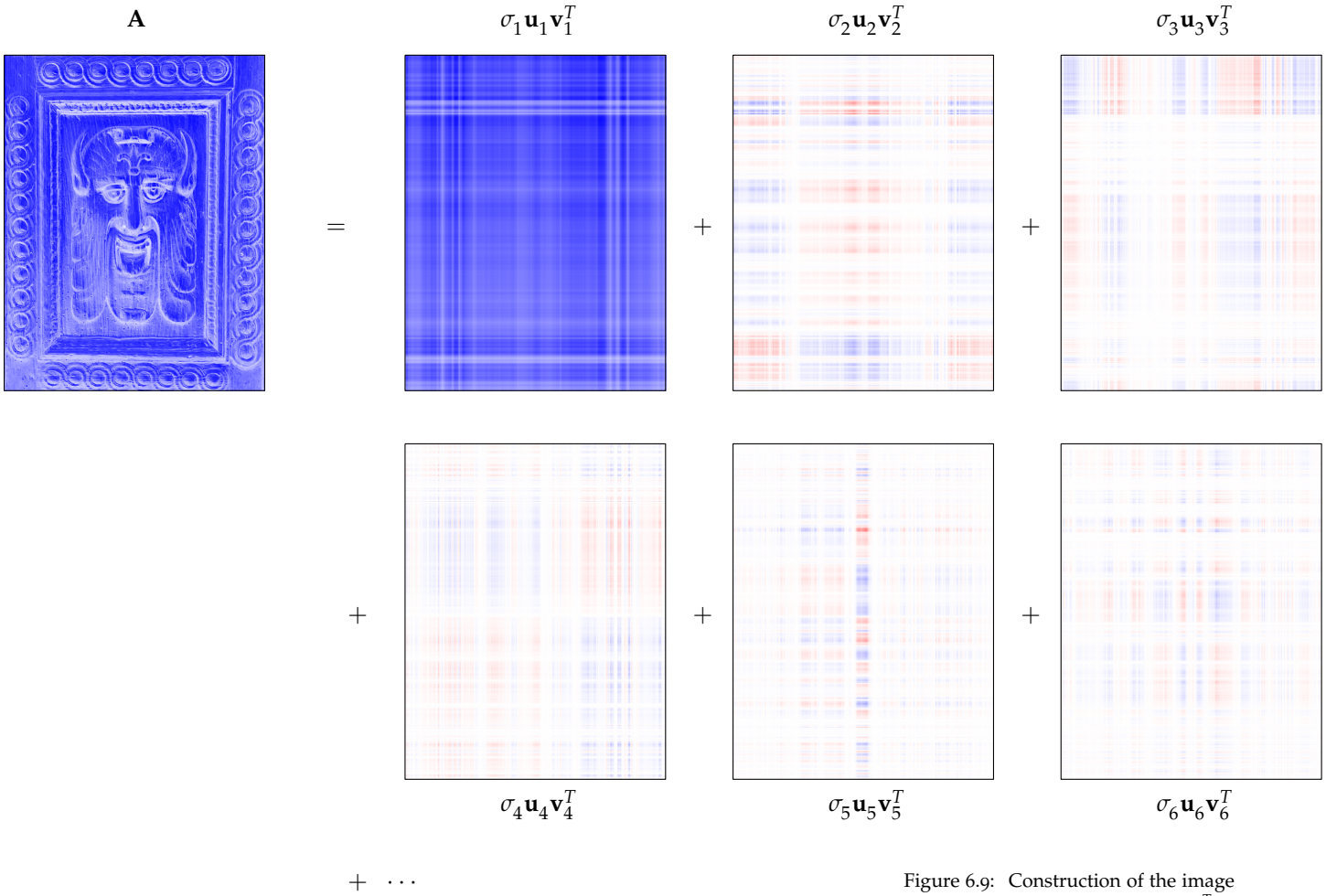


Figure 6.9: Construction of the image $\mathbf{A}$ from the individual terms $\sigma_j \mathbf{u}_j \mathbf{v}_j^T$. The first term makes a major contribution; each subsequent term makes a small adjustment, and these adjustments diminish in significance as j increases.

6.12 Principal Component Analysis

Linear algebra enables the analysis of the volumes of data that now so commonly arise from applications ranging from basic science to public policy. Such measured data often depends on many factors, and we seek to identify those that are most critical. Within this realm of multivariate statistics, *principal component analysis* (PCA) is a fundamental tool.

Linear algebraists often say, "PCA is the SVD" – in this section, we will explain what this means, and some of the subtleties involved.

6.12.1 Variance and covariance

To understand principal component analysis, we need some basic notions from statistics, described in any basic textbook. For a general description of PCA along with numerous applications, see the text by Jolliffe [2], whose presentation shaped parts of our discussion here.

The *expected value*, or *mean*, of a random variable X is denoted $\mathsf{E}[X]$. The expected value is a linear function, so for any constants $\alpha, \beta \in \mathbb{R}$, $\mathsf{E}[\alpha X + \beta] = \alpha \mathsf{E}[X] + \beta$.

The *variance* of X describes how much X is expected to deviate from its mean,

$$\mathsf{Var}(X) = \mathsf{E}[(X - \mathsf{E}[X])^2],$$

which, using linearity of the expected value, takes the equivalent form

$$\mathsf{Var}(X) = \mathsf{E}[X^2] - \mathsf{E}[X]^2.$$

The *covariance* between two (potentially correlated) random variables X and Y is

$$\begin{aligned} \mathsf{Cov}(X, Y) &= \mathsf{E}[(X - \mathsf{E}[X])(Y - \mathsf{E}[Y])] \\ &= \mathsf{E}[XY] - \mathsf{E}[X]\mathsf{E}[Y]. \end{aligned}$$

with $\mathsf{Cov}(X, X) = \mathsf{Var}(X)$. These definitions of variance and covariance are the bedrock concepts underneath PCA, for with them we can understand the variance present in a linear combination of several random variables.

Suppose we have a set of real-valued random variables $X_1, \ldots, X_n$ in which we suspect there may be some redundancy. Perhaps some of these variables can be expressed as linear combinations of the others – either exactly, or nearly so. At the other extreme, there may be some way to combine $X_1, \ldots, X_n$ that captures much of the variance in one (or a few) aggregate random variables. In particular, we shall seek scalars $\gamma_1, \ldots, \gamma_n$ such that

$$\sum_{j=1}^{n} \gamma_j X_j$$

has the largest possible variance. The definitions of variance and covariance, along with the linearity of the expected value, lead to a formula for the variance of a linear combination of random variables:

$$\mathsf{Var}\Big(\sum_{j=1}^{n}\gamma_j X_j\Big) = \sum_{j=1}^{n}\sum_{k=1}^{n}\gamma_j\gamma_k\,\mathsf{Cov}(X_j, X_k). \tag{6.14}$$

You have seen double sums like this before. If we define the *covariance matrix* $\mathbf{C} \in \mathbb{R}^{n\times n}$ having (j,k) entry

$$c_{j,k} = \mathsf{Cov}(X_j, X_k),$$

and let $\mathbf{v} = [\gamma_1, \ldots, \gamma_n]^T$, then the variance of the combined variable is just a Rayleigh quotient:

$$\mathsf{Var}\Big(\sum_{j=1}^{n}\gamma_j X_j\Big) = \mathbf{v}^T\mathbf{C}\mathbf{v}.$$

Since the covariance function is symmetric: $\mathsf{Cov}(X,Y) = \mathsf{Cov}(Y,X)$, the matrix $\mathbf{C}$ is symmetric; it is also positive semidefinite. Why? Variance, by its definition as the expected value of the square of a real random variable, is always nonnegative. Thus the formula (6.14), which derives from the linearity of the expected value, ensures that $\mathbf{v}^T\mathbf{C}\mathbf{v} \geq 0$. (Under what circumstances can this quantity be zero?)

We can write $\mathbf{C}$ in another convenient way. Collect the random variables into the vector

$$\mathbf{X} = \begin{bmatrix} X_1 \\ \vdots \\ X_n \end{bmatrix}.$$

Then the (j,k) entry of $\mathsf{E}[\mathbf{X}\mathbf{X}^T] - \mathsf{E}[\mathbf{X}]\mathsf{E}[\mathbf{X}]^T$ is

$$\mathsf{E}[X_jX_k] - \mathsf{E}[X_j]\mathsf{E}[X_k] = \mathsf{Cov}(X_j, X_k) = c_{j,k},$$

and so

$$\mathbf{C} = \mathsf{E}[\mathbf{X}\mathbf{X}^T] - \mathsf{E}[\mathbf{X}]\mathsf{E}[\mathbf{X}]^T.$$

6.12.2 *Derived variables that maximize variance*

Return now to the problem of *maximizing* the variance of $\mathbf{v}^T\mathbf{C}\mathbf{v}$. Without constraint on $\mathbf{v}$, this quantity can be arbitrarily large (assuming $\mathbf{C}$ is nonzero); thus we shall require that $\sum_{j=1}^{k}\gamma_j^2 = \|\mathbf{v}\|^2 = 1$. With this normalization, you immediately see how to maximize the variance $\mathbf{v}^T\mathbf{C}\mathbf{v}$: $\mathbf{v}$ should be a unit eigenvector associated with the largest magnitude eigenvalue of $\mathbf{C}$; call this vector $\mathbf{v}_1$. The associated variance, of course, is the largest eigenvalue of $\mathbf{C}$; call it

$$\lambda_1 = \mathbf{v}_1^T\mathbf{C}\mathbf{v}_1 = \max_{\mathbf{v}\in\mathbb{C}^n}\frac{\mathbf{v}^T\mathbf{C}\mathbf{v}}{\mathbf{v}^T\mathbf{v}}.$$

The eigenvector $\mathbf{v}_1$ encodes the way to combine $X_1, \ldots, X_n$ to maximize variance. The new variable – *the leading principal component* – is

$$\mathbf{v}_1^T \mathbf{X} = \sum_{j=1}^{n} \gamma_j X_j.$$

You are already suspecting that a unit eigenvector associated with the second largest eigenvalue, $\mathbf{v}_2$ with $\lambda_2 = \mathbf{v}_2^T \mathbf{C} \mathbf{v}_2$, must encode the second-largest way to maximize variance.

Let us explore this intuition. To find the second-best way to combine the variables, we should insist that the next new variable, for now call it $\mathbf{w}^T \mathbf{X}$, should be *uncorrelated* with the first, i.e.,

$$\mathsf{Cov}(\mathbf{v}_1^T \mathbf{X}, \mathbf{w}^T \mathbf{X}) = 0.$$

However, using linearity of expectation and the fact that, e.g., $\mathbf{w}^T \mathbf{X} = \mathbf{X}^T \mathbf{w}$ for real vectors,

$$\begin{aligned}
\mathsf{Cov}(\mathbf{v}_1^T \mathbf{X}, \mathbf{w}^T \mathbf{X}) &= \mathsf{E}[(\mathbf{v}_1^T \mathbf{X})(\mathbf{w}^T \mathbf{X})] - \mathsf{E}[\mathbf{v}_1^T \mathbf{X}]\mathsf{E}[\mathbf{w}^T \mathbf{X}] \\
&= \mathsf{E}[(\mathbf{v}_1^T \mathbf{X}\mathbf{X}^T \mathbf{w}] - \mathsf{E}[\mathbf{v}_1^T \mathbf{X}]\mathsf{E}[\mathbf{X}^T \mathbf{w}] \\
&= \mathbf{v}_1^T \mathsf{E}[\mathbf{X}\mathbf{X}^T]\mathbf{w} - \mathbf{v}_1^T \mathsf{E}[\mathbf{X}]\mathsf{E}[\mathbf{X}^T]\mathbf{w} \\
&= \mathbf{v}_1^T \left(\mathsf{E}[\mathbf{X}\mathbf{X}^T] - \mathsf{E}[\mathbf{X}]\mathsf{E}[\mathbf{X}^T]\right)\mathbf{w} \\
&= \mathbf{v}_1^T \mathbf{C} \mathbf{w} = \lambda_1 \mathbf{v}_1^T \mathbf{w}.
\end{aligned}$$

Hence (assuming $\lambda_1 \neq 0$), for the combined variables $\mathbf{v}_1^T \mathbf{X}$ and $\mathbf{w}^T \mathbf{X}$ to be uncorrelated, the vectors $\mathbf{v}_1$ and $\mathbf{w}$ must be *orthogonal*, perfectly confirming your intuition: the second-best way to combine the variables is to pick $\mathbf{w}$ to be a unit eigenvector $\mathbf{v}_2$ of $\mathbf{C}$ corresponding to the second largest eigenvalue. Since the eigenvectors of a symmetric matrix are orthogonal, we optimize over all vectors orthogonal to $\mathbf{u}_1$. The associated variance of $\mathbf{v}_2^T \mathbf{X}$ is thus

$$\lambda_2 = \max_{\mathbf{w} \perp \mathrm{span}\{\mathbf{v}_1\}} \frac{\mathbf{w}^T \mathbf{C} \mathbf{w}}{\mathbf{w}^T \mathbf{w}}.$$

Of course, in general, the kth best way to combine the variables is given by the eigenvector $\mathbf{v}_k$ of $\mathbf{C}$ associated with the kth largest eigenvalue.

We learn much about our variables from the relative size of the variances (eigenvalues)

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n \geq 0.$$

If some of the latter eigenvalues are very small, that indicates that the set of n random variables can be well approximated by a fewer number of aggregate variables. These aggregate variables are the *principal components* of $X_1, \ldots, X_n$.

6.12.3 *Approximate PCA from empirical data*

In practical situations, one often seeks to analyze empirical data drawn from some unknown distribution: the expected values and covariances are not available. Instead, we will *estimate* these from the measured data.

Suppose, as before, that we are considering n random variables, $X_1, \ldots, X_n$, with m samples of each:

$$x_{j,k}, \quad k = 1, \ldots, m,$$

i.e., $x_{j,k}$ is the kth sample of the random variable X_j. The expected value has the familiar *unbiased estimate*

$$\mu_j = \frac{1}{m} \sum_{k=1}^{m} x_{j,k}.$$

Similarly, we can approximate the covariance

$$\mathsf{Cov}(X_j, X_k) = \mathsf{E}[(X_j - \mathsf{E}[X_j])(X_k - \mathsf{E}[X_k)].$$

One might naturally estimate this as

$$\frac{1}{m} \sum_{\ell=1}^{m} (x_{j,\ell} - \mu_j)(x_{k,\ell} - \mu_k).$$

However, replacing the true expected values $\mathsf{E}[X_j]$ and $\mathsf{E}[X_k]$ with the empirical estimates μ_j and μ_k introduces some slight bias into this estimate. This bias can be removed by scaling, replacing $1/m$ by $1/(m-1)$ to get the *unbiased estimate*

$$s_{j,k} = \frac{1}{m-1} \sum_{\ell=1}^{m} (x_{j,\ell} - \mu_j)(x_{k,\ell} - \mu_k), \quad j, k = 1, \ldots, n.$$

If we let

$$\mathbf{x}_j = \begin{bmatrix} x_{j,1} \\ \vdots \\ x_{j,m} \end{bmatrix}, \quad j = 1, \ldots, n,$$

then each covariance estimate is just an inner product

$$s_{j,k} = \frac{1}{m-1} (\mathbf{x}_j - \mu_j)^T (\mathbf{x}_k - \mu_k).$$

Here the notation $\mathbf{x}_j - \mu_j$ means: subtract the scalar μ_j from all entries of the vector $\mathbf{x}_j$.

Thus, if we center the samples of each variable about its empirical mean, we can write the empirical covariance matrix $\mathbf{S} = [s_{j,k}]$ as a matrix product. Let

$$\mathfrak{X} := [\, (\mathbf{x}_1 - \mu_1) \quad (\mathbf{x}_2 - \mu_2) \quad \cdots \quad (\mathbf{x}_n - \mu_n) \,] \in \mathbb{R}^{m \times n},$$

so that

$$\mathbf{S} = \frac{1}{m-1} \mathfrak{X}^T \mathfrak{X}. \tag{6.15}$$

Now conduct principal component analysis just as before, but with the empirical covariance matrix $\mathbf{S}$ replacing the true covariance matrix $\mathbf{C}$. The eigenvectors of $\mathbf{S}$ now lead to *sample principal components*.

Where is the connection to the singular value decomposition? Notice how we formed the sample covariance matrix $\mathbf{S}$ in equation (6.15). Aside from the scaling $1/(m-1)$, this structure recalls the first step in our construction of the singular value decomposition earlier in the chapter. We can thus arrive at the sample principal components by computing the singular value decomposition of the data matrix $\mathfrak{X}$. This is why some say, "PCA is just the SVD." We summarize the details step-by-step.

To compute the singular value decomposition of some matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$, start by computing the eigenvalues and eigenvectors of $\mathbf{A}^T\mathbf{A}$. In our setting, the eigenvectors of $\mathbf{S}$ are the right singular vectors of $\mathfrak{X}$.

1. Collect m samples of each of n random variables, $x_{j,k}$ for $j = 1, \ldots, n$ and $k = 1, \ldots, m$. (We need $m > 1$; typically $m \gg n$.)
2. Compute the empirical means of each column, $\mu_k = (\sum_{k=1}^{m} x_{j,k})/m$.
3. Stacking the samples of the kth variable in the vector $\mathbf{x}_k \in \mathbb{R}^m$, construct the mean-centered data matrix

$$\mathfrak{X} = [\,(\mathbf{x}_1 - \mu_1) \quad (\mathbf{x}_2 - \mu_2) \quad \cdots \quad (\mathbf{x}_n - \mu_n)\,] \in \mathbb{R}^{m \times n}.$$

4. Compute the (skinny) singular value decomposition $\mathfrak{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$, with $\mathbf{U} \in \mathbb{R}^{m \times n}$, $\mathbf{\Sigma} = \operatorname{diag}(\sigma_1, \ldots, \sigma_n) \in \mathbb{R}^{n \times n}$, and $\mathbf{V} = [\mathbf{v}_1 \ \cdots \ \mathbf{v}_n] \in \mathbb{R}^{n \times n}$.
5. The kth sample principal component is given by $\mathbf{v}_k^T\mathbf{X}$, where $\mathbf{X} = [X_1, \ldots, X_n]^T$ is the vector of random variables.
6. You can assess the importance of the various principal components via the eigenvalues of $\mathbf{S}$, given by $\lambda_k = \sigma_k^2/(m-1)$. If these eigenvalues decay rapidly as k increases, that is a sign that your data can be well-represented by the first few principal components.

A word of caution: when conducting principal component analysis, the *scale* of each column matters. For example, if the random variables sampled in each column of $\mathfrak{X}$ are measurements of physical quantities, they can differ considerably in magnitude depending on the units of measurement. By changing units of measurement, you can significantly alter the principal components.

6.12.4 Clustering via PCA

PCA can be used to cluster data. To illustrate, we turn to a data set comprising of chemical properties of Italian wines. The data set includes measurements of 13 different properties for 178 wines, giving

a data matrix $\mathfrak{X}$ of dimension 178×13. (The properties include: alcohol content, malic acid, ash, alcalinity of the ash, etc.)

Each of these 178 wines comes from one of three grape varieties: Barolo (nebbiolo grape), Grignolino, or Barbera. Now a bottle of Barolo typically costs quite a bit more than the other two wines, so it would be interesting to know if these high-end wines really can be distinguished, chemically, from the others.

When working with a real data set, we must begin by preparing the data. Since variables may be measured in different units, we begin by computing the empirical mean of each variable, and dividing by the mean (so that each variable now has mean 1).

With this normalization complete, conduct PCA as described above: form the data matrix $\mathfrak{X}$ and compute its dominant singular values and singular vectors. Figure 6.10 shows the singular values of $\mathfrak{X}$, suggesting that the first two or three principal components will dominate the others.

How can we use principal components to *cluster* the data? Consider the kth sample of data, described by the variables

$$x_{1,k}, \quad x_{2,k}, \quad , \ldots , \quad , x_{13,k}.$$

Denote the first right singular vector of $\mathfrak{X} \in \mathbb{R}^{178\times 13}$ by

$$\mathbf{v}_1 = [\gamma_1, \ldots, \gamma_{13}]^T.$$

Then the variance-maximizing combination of the 13 variables is given by

$$\xi_k := \sum_{j=1}^{13} \gamma_j x_{j,k}.$$

Similarly, writing the second right singular vector of $\mathfrak{X}$ as

$$\mathbf{v}_2 = [\omega, \ldots, \omega_{13}]^T,$$

You can download the "Wine Data Set" from the UCI Machine Learning Repository, `https://archive.ics.uci.edu/ml/datasets/wine`. For more details on this data set and the application of eigenvector-based clustering to it, see: M. Forina, C. Armanino, M. Castino, and M. Ubigli. "Multivariate data analysis as a discriminating method of the origin of wines," *Vitis* 25 (1986) 189–201.

If you do not normalize your variables, you risk having an extremely large principal component dominated by one single variable that happens to have very large values.

Using $\lambda_k = \sigma_k^2/(m-1)$, we have

$$\lambda_1 \approx 0.608, \qquad \lambda_2 \approx 0.289,$$
$$\lambda_3 \approx 0.161, \qquad \lambda_4 \approx 0.089.$$

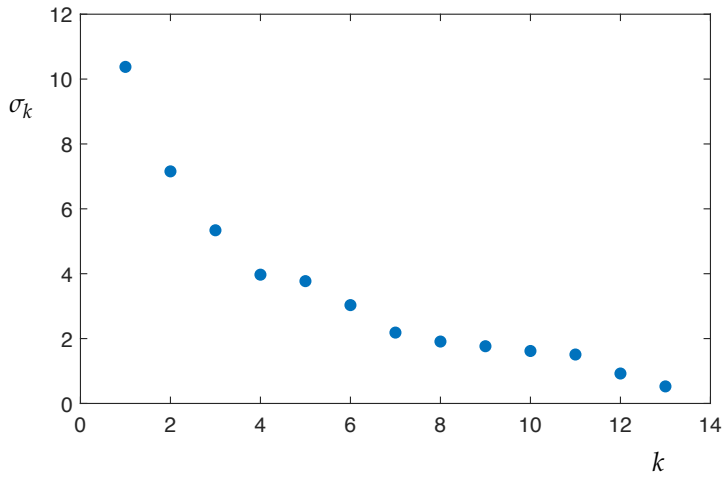


Figure 6.10: Singular values of the 178×13 wine data matrix $\mathfrak{X}$ (with normalized columns).

define the second-best variance maximizing combination as

$$\eta_k := \sum_{j=1}^{13} \omega_j x_{j,k}.$$

Here is the key idea: we have squeezed as much variance as possible from our 13 variables into 2 variables. Can we actually *reduce the dimension* of our data set from those 13 variables down into the two new variables?

$$(x_{1,k}, x_{2,k}, \ldots, x_{13,k}) \quad \Longrightarrow \quad (\xi_k, \eta_k)$$

If $\lambda_3 = \lambda_4 = \cdots = \lambda_{13} = 0$, then this would be a perfect compression of the variables. Of course in practice, the reduction is only approximate, but hopefully we have distilled the essential distinguishing features of the 13 variables into those 2 consolidated variables ξ_k and η_k. We can view (ξ_k, η_k) as a *projection* of the 13 dimensional data onto a two-dimensional space. Many other projections are possible (just take any two of the given variables), but the one from PCA is optimal (in the sense of maximizing variance). Figure 6.11 shows the (ξ_k, η_k) projection for these 178 data points.

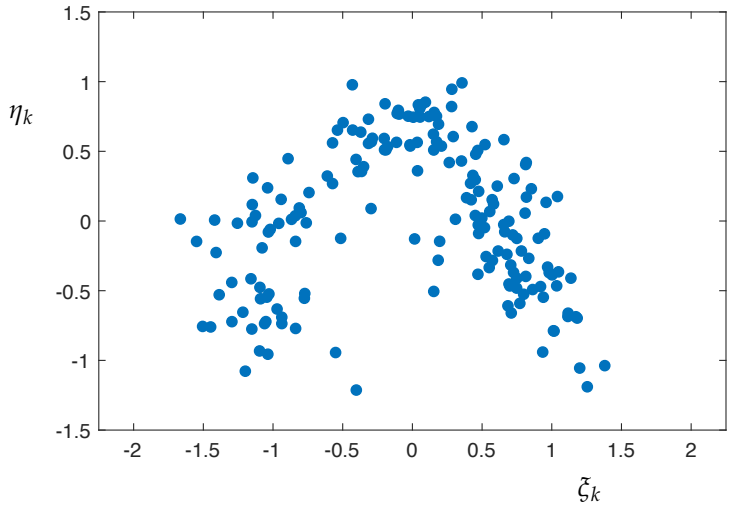


Figure 6.11: Projection of the wine data set into the two variables defined by the leading two principal components.

Recall that our goal is to identify if each of these samples is Barolo, Grignolino, or Barbera. Can you see any clusters in Figure 6.11? To help, we apply the k-means algorithm with $k = 3$ to this data. Figure 6.12 shows the results.

We used MATLAB's `kmeans` implementation, running from 10 starting configurations and keeping the best clustering that results.

Conveniently enough, we have labeled data in this case, so we can check if the clustering in Figure 6.12 did a good job of identifying the three wine varieties. Figure 6.13 shows the results.

First of all, we notice that the three wine varieties really do look quite distinct, when projected into the two-dimensional (ξ_k, η_k) PCA

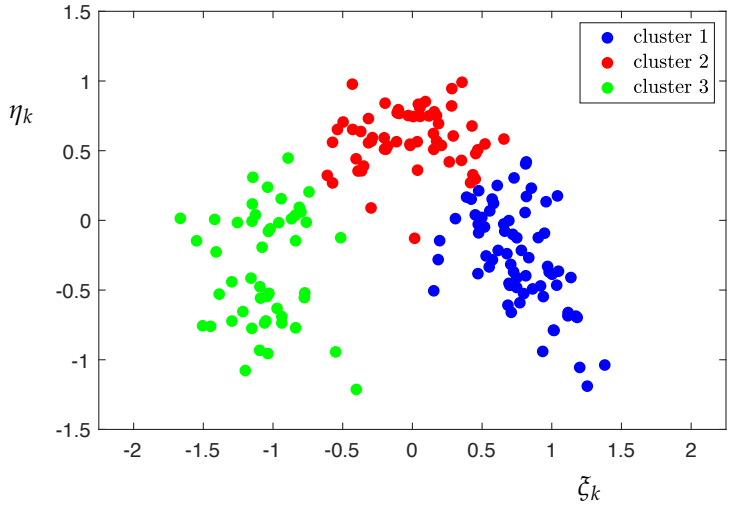


Figure 6.12: Results of k-means clustering of the results in Figure 6.11.

coordinates. Even better, the k-means results match these pretty well: k-means made a few mistakes, especially at the frontier between Barolo and Grignolino, but overall it looks like we could do a decent job of identifying wine through the combined efforts of PCA and k-means. (Whether, for this application, data science yields an improvement over the traditional manner of careful testing with a well-trained palate, I will let you be the judge. . . .)

Indeed, k-means draws a cleaner boundary between these wines than we see in reality.

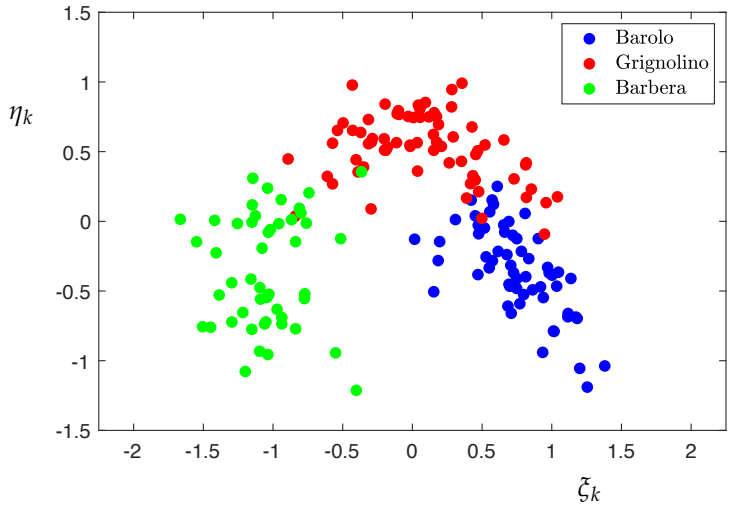


Figure 6.13: Repetition of Figure 6.11, but now color-coded according to the wine variety specified in the data set.

The derivation above suggests that the PCA-derived variables (ξ_k, η_k) should exhibit greater variance than we would find from just picking a pair of the variables (say, $(x_{1,k}, x_{2,k})$. Figure 6.14 verifies this intuition, projecting the 13-dimensional data onto just two coordinates (we picked (1,2), (3,4), and (5,6)). (The color-coding refers to the

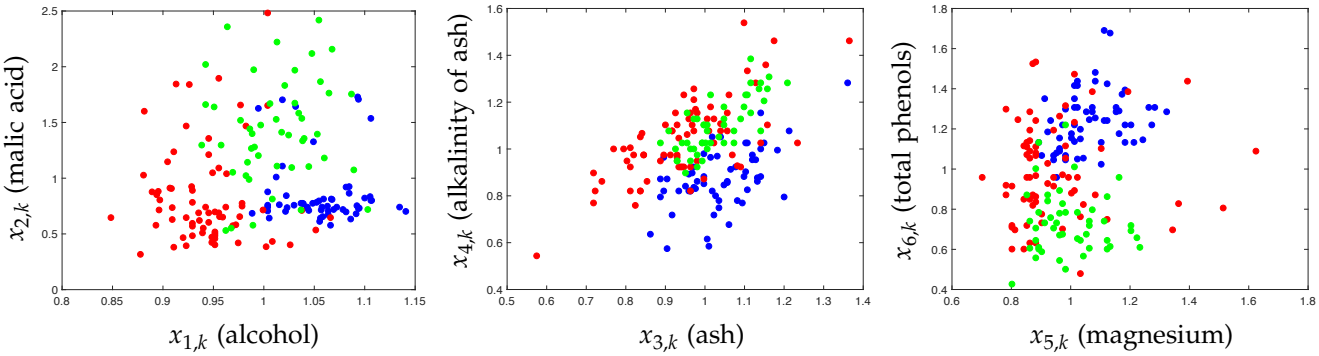


Figure 6.14: These three plots each project the 13-dimensional wine data set into two dimensions, just using the given variables instead of the derived variables from PCA. We see that the given variables do a poor job of distinguishing the three classes of wines.

true wine varieties, as in Figure 6.13.)

6.13 Afterword

The singular value decomposition was developed in its initial form by Eugenio Beltrami (1873) and, independently, by Camille Jordan (1874).[3]

[3]