Singular Value Decomposition

CONTENTS

CHAPTER 6 derived a number of algorithms for computing the eigenvalues and eigenvectors of matrices $A \in \mathbb{R}^{n\times n}$. Using this machinery, we complete our initial discussion of numerical linear algebra by deriving and making use of one final matrix factorization that exists for *any* matrix $A \in \mathbb{R}^{m\times n}$, even if it is not symmetric or square: the singular value decomposition (SVD).

7.1 DERIVING THE SVD

For $A \in \mathbb{R}^{m\times n}$, we can think of the function $\vec{v} \mapsto A\vec{v}$ as a map taking points $\vec{v} \in \mathbb{R}^n$ to points $A\vec{v} \in \mathbb{R}^m$. From this perspective, we might ask what happens to the geometry of $\mathbb{R}^n$ in the process, and in particular the effect A has on lengths of and angles between vectors.

Applying our usual starting point for eigenvalue problems, we examine the effect that A has on the lengths of vectors by examining critical points of the ratio

$$R(\vec{v}) = \frac{\|A\vec{v}\|_2}{\|\vec{v}\|_2}$$

over various vectors $\vec{v} \in \mathbb{R}^n\backslash\{\vec{0}\}$. This quotient measures relative shrinkage or growth of $\vec{v}$ under the action of A. Scaling $\vec{v}$ does not matter, since

$$R(\alpha\vec{v}) = \frac{\|A\cdot\alpha\vec{v}\|_2}{\|\alpha\vec{v}\|_2} = \frac{|\alpha|}{|\alpha|}\cdot\frac{\|A\vec{v}\|_2}{\|\vec{v}\|_2} = \frac{\|A\vec{v}\|_2}{\|\vec{v}\|_2} = R(\vec{v}).$$

Thus, we can restrict our search to $\vec{v}$ with $\|\vec{v}\|_2 = 1$. Furthermore, since $R(\vec{v}) \geq 0$, we can instead find critical points of $[R(\vec{v})]^2 = \|A\vec{v}\|_2^2 = \vec{v}^\top A^\top A\vec{v}$. As we have shown in previous chapters, critical points of $\vec{v}^\top A^\top A\vec{v}$ subject to $\|\vec{v}\|_2 = 1$ are exactly the eigenvectors $\vec{v}_i$ satisfying $A^\top A\vec{v}_i = \lambda_i\vec{v}_i$; we know $\lambda_i \geq 0$ and $\vec{v}_i \cdot \vec{v}_j = 0$ when $i \neq j$ since $A^\top A$ is symmetric and positive semidefinite.

Based on our use of the function R, the $\{\vec{v}_i\}$ basis is a reasonable one for studying the effects of A on $\mathbb{R}^n$. Returning to the original goal of characterizing the action of A from a geometric standpoint, define $\vec{u}_i \equiv A\vec{v}_i$. We can make an observation about $\vec{u}_i$ revealing a second eigenvalue structure:

$$\begin{aligned}
\lambda_i\vec{u}_i &= \lambda_i \cdot A\vec{v}_i \text{ by definition of } \vec{u}_i \\
&= A(\lambda_i\vec{v}_i) \\
&= A(A^\top A\vec{v}_i) \text{ since } \vec{v}_i \text{ is an eigenvector of } A^\top A \\
&= (AA^\top)(A\vec{v}_i) \text{ by associativity} \\
&= (AA^\top)\vec{u}_i.
\end{aligned}$$

Taking norms shows $\|\vec{u}_i\|_2 = \|A\vec{v}_i\|_2 = \sqrt{\|A\vec{v}_i\|_2^2} = \sqrt{\vec{v}_i^\top A^\top A\vec{v}_i} = \sqrt{\lambda_i}\|\vec{v}_i\|_2$. This formula leads to one of two conclusions:

1. Suppose $\vec{u}_i \neq \vec{0}$. In this case, $\vec{u}_i = A\vec{v}_i$ is a *corresponding* eigenvector of $AA^\top$ with $\|\vec{u}_i\|_2 = \sqrt{\lambda_i}\|\vec{v}_i\|_2$.
2. Otherwise, $\vec{u}_i = \vec{0}$.

An identical proof shows that if $\vec{u}$ is an eigenvector of $AA^\top$, then $\vec{v} \equiv A^\top\vec{u}$ is either zero or an eigenvector of $A^\top A$ with the same eigenvalue.

Take k to be the number of strictly positive eigenvalues $\lambda_i > 0$ for $i \in \{1, \ldots, k\}$. By our construction above, we can take $\vec{v}_1, \ldots, \vec{v}_k \in \mathbb{R}^n$ to be eigenvectors of $A^\top A$ and corresponding eigenvectors $\vec{u}_1, \ldots, \vec{u}_k \in \mathbb{R}^m$ of $AA^\top$ such that

$$\begin{aligned}
A^\top A\vec{v}_i &= \lambda_i\vec{v}_i \\
AA^\top\vec{u}_i &= \lambda_i\vec{u}_i
\end{aligned}$$

for eigenvalues $\lambda_i > 0$; here, we normalize such that $\|\vec{v}_i\|_2 = \|\vec{u}_i\|_2 = 1$ for all i. Define matrices $\bar{V} \in \mathbb{R}^{n\times k}$ and $\bar{U} \in \mathbb{R}^{m\times k}$ whose columns are $\vec{v}_i$'s and $\vec{u}_i$'s, respectively. By construction, $\bar{U}$ contains an orthogonal basis for the column space of A, and $\bar{V}$ contains an orthogonal basis for the row space of A.

We can examine the effect of these new basis matrices on A. Take $\vec{e}_i$ to be the i-th standard basis vector. Then,

$$\begin{aligned}
\bar{U}^\top A\bar{V}\vec{e}_i &= \bar{U}^\top A\vec{v}_i \text{ by definition of } \bar{V} \\
&= \frac{1}{\lambda_i}\bar{U}^\top A(\lambda_i\vec{v}_i) \text{ since we assumed } \lambda_i > 0 \\
&= \frac{1}{\lambda_i}\bar{U}^\top A(A^\top A\vec{v}_i) \text{ since } \vec{v}_i \text{ is an eigenvector of } A^\top A \\
&= \frac{1}{\lambda_i}\bar{U}^\top(AA^\top)A\vec{v}_i \text{ by associativity} \\
&= \frac{1}{\sqrt{\lambda_i}}\bar{U}^\top(AA^\top)\vec{u}_i \text{ since we rescaled so that } \|\vec{u}_i\|_2 = 1 \\
&= \sqrt{\lambda_i}\bar{U}^\top\vec{u}_i \text{ since } AA^\top\vec{u}_i = \lambda_i\vec{u}_i \\
&= \sqrt{\lambda_i}\vec{e}_i.
\end{aligned}$$

Take $\bar{\Sigma} = \text{diag}(\sqrt{\lambda_1}, \ldots, \sqrt{\lambda_k})$. Then, the derivation above shows that $\bar{U}^\top A\bar{V} = \bar{\Sigma}$.

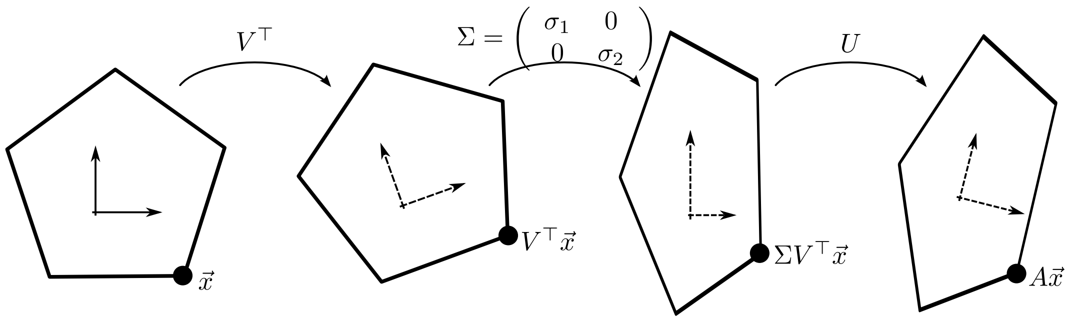


Figure 7.1 Geometric interpretation for the singular value decomposition $A = U\Sigma V^\top$. The matrices U and $V^\top$ are orthogonal and hence preserve lengths and angles. The diagonal matrix Σ scales the horizontal and vertical axes independently.

Complete the columns of $\bar{U}$ and $\bar{V}$ to $U \in \mathbb{R}^{m\times m}$ and $V \in \mathbb{R}^{n\times n}$ by adding orthonormal null space vectors $\vec{v}_i$ and $\vec{u}_i$ with $A^\top A\vec{v}_i = \vec{0}$ and $AA^\top\vec{u}_i = \vec{0}$, respectively. After this extension, $U^\top AV\vec{e}_i = \vec{0}$ and/or $\vec{e}_i^{\,\top} U^\top AV = \vec{0}^\top$ for $i > k$. If we take

$$\Sigma_{ij} \equiv \begin{cases} \sqrt{\lambda_i} & i = j \text{ and } i \leq k \\ 0 & \text{otherwise,} \end{cases}$$

then we can extend our previous relationship to show $U^\top AV = \Sigma$, or, by orthogonality of U and V,

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

This factorization is the *singular value decomposition* (SVD) of A. The columns of U are called the *left singular vectors*, and the columns of V are called the *right singular vectors*. The diagonal elements σ_i of Σ are the *singular values* of A; usually they are sorted such that $\sigma_1 \geq \sigma_2 \geq \cdots \geq 0$. Both U and V are orthogonal matrices; the columns of U and V corresponding to $\sigma_i \neq 0$ span the column and row spaces of A, respectively.

The SVD provides a complete geometric characterization of the action of A. Since U and V are orthogonal, they have no effect on lengths and angles; as a diagonal matrix, Σ scales individual coordinate axes. Since the SVD always exists, *all* matrices $A \in \mathbb{R}^{m\times n}$ are a composition of an isometry, a scale in each coordinate, and a second isometry. This sequence of operations is illustrated in Figure 7.1.

7.1.1 Computing the SVD

The columns of V are the eigenvectors of $A^\top A$, so they can be computed using algorithms discussed in the previous chapter. Rewriting $A = U\Sigma V^\top$ as $AV = U\Sigma$, the columns of U corresponding to nonzero singular values in Σ are normalized columns of AV. The remaining columns satisfy $AA^\top\vec{u}_i = \vec{0}$ and can be computed using the LU factorization.

This is by no means the most efficient or stable way to compute the SVD, but it works reasonably well for many applications. We omit more specialized algorithms for finding the SVD, but many of them are extensions of power iteration and other algorithms we already have covered that avoid forming $A^\top A$ or $AA^\top$ explicitly.

7.2 APPLICATIONS OF THE SVD

We devote the remainder of this chapter to introducing applications of the SVD. The SVD appears countless times in both the theory and practice of numerical linear algebra, and its importance hardly can be exaggerated.

7.2.1 Solving Linear Systems and the Pseudoinverse

In the special case where $A \in \mathbb{R}^{n\times n}$ is square and invertible, the SVD can be used to solve the linear problem $A\vec{x} = \vec{b}$. By substituting $A = U\Sigma V^\top$, we have $U\Sigma V^\top \vec{x} = \vec{b}$, or by orthogonality of U and V,

$$\vec{x} = V\Sigma^{-1}U^\top \vec{b}.$$

Σ is a square diagonal matrix, so Σ^{-1} is the matrix with diagonal entries $1/\sigma_i$.

Computing the SVD is far more expensive than most of the linear solution techniques we introduced in Chapter 3, so this initial observation mostly is of theoretical rather than practical interest. More generally, however, suppose we wish to find a least-squares solution to $A\vec{x} \approx \vec{b}$, where $A \in \mathbb{R}^{m\times n}$ is not necessarily square. From our discussion of the normal equations, we know that $\vec{x}$ must satisfy $A^\top A\vec{x} = A^\top \vec{b}$. But when A is "short" or "underdetermined," that is, when A has more columns than rows ($m < n$) or has linearly dependent columns, the solution to the normal equations might not be unique.

To cover the under-, completely-, and overdetermined cases simultaneously without resorting to regularization (see §4.1.3), we can solve an optimization problem of the following form:

$$\begin{aligned} \text{minimize} \quad & \|\vec{x}\|_2^2 \\ \text{subject to} \quad & A^\top A\vec{x} = A^\top \vec{b}. \end{aligned}$$

This optimization chooses the vector $\vec{x} \in \mathbb{R}^n$ with least norm that satisfies the normal equations $A^\top A\vec{x} = A^\top \vec{b}$. When $A^\top A$ is invertible, meaning the least-squares problem is completely- or overdetermined, there is only one $\vec{x}$ satisfying the constraint. Otherwise, of all the feasible vectors $\vec{x}$, we choose the one with minimal $\|\vec{x}\|_2$. That is, we seek the smallest possible least-square solution of $A\vec{x} \approx \vec{b}$, when multiple $\vec{x}$'s minimize $\|A\vec{x} - \vec{b}\|_2$.

Write $A = U\Sigma V^\top$. Then,

$$\begin{aligned} A^\top A &= (U\Sigma V^\top)^\top (U\Sigma V^\top) \\ &= V\Sigma^\top U^\top U\Sigma V^\top \text{ since } (AB)^\top = B^\top A^\top \\ &= V\Sigma^\top \Sigma V^\top \text{ since } U \text{ is orthogonal.} \end{aligned}$$

Using this expression, the constraint $A^\top A\vec{x} = A^\top \vec{b}$ can be written

$$\begin{aligned} V\Sigma^\top \Sigma V^\top \vec{x} &= V\Sigma^\top U^\top \vec{b}, \\ \text{or equivalently, } \Sigma^\top \Sigma \vec{y} &= \Sigma^\top \vec{d}, \end{aligned}$$

after taking $\vec{d} \equiv U^\top \vec{b}$ and $\vec{y} \equiv V^\top \vec{x}$.

Since V is orthogonal, $\|\vec{y}\|_2 = \|\vec{x}\|_2$ and our optimization becomes:

$$\begin{aligned} \text{minimize} \quad & \|\vec{y}\|_2^2 \\ \text{subject to} \quad & \Sigma^\top \Sigma \vec{y} = \Sigma^\top \vec{d}. \end{aligned}$$

Since Σ is diagonal, the condition $\Sigma^\top \Sigma \vec{y} = \Sigma^\top \vec{d}$ can be written $\sigma_i^2 y_i = \sigma_i d_i$. So, whenever $\sigma_i \neq 0$ we must have $y_i = d_i/\sigma_i$. When $\sigma_i = 0$, there is *no* constraint on y_i. Since we are minimizing $\|\vec{y}\|_2^2$ we might as well take $y_i = 0$. In other words, the solution to this optimization is $\vec{y} = \Sigma^+\vec{d}$, where $\Sigma^+ \in \mathbb{R}^{n\times m}$ has the form:

$$\Sigma^+_{ij} \equiv \begin{cases} 1/\sigma_i & i = j \text{ and } \sigma_i \neq 0 \\ 0 & \text{otherwise.} \end{cases}$$

Undoing the change of variables, this result in turn yields $\vec{x} = V\vec{y} = V\Sigma^+\vec{d} = V\Sigma^+U^\top\vec{b}$.

With this motivation, we make the following definition:

Definition 7.1 (Pseudoinverse)**.** The *pseudoinverse* of $A = U\Sigma V^\top \in \mathbb{R}^{m\times n}$ is $A^+ \equiv V\Sigma^+U^\top \in \mathbb{R}^{n\times m}$.

Our derivation above shows that the pseudoinverse of A enjoys the following properties:

- When A is square and invertible, $A^+ = A^{-1}$.
- When A is overdetermined, $A^+\vec{b}$ gives the least-squares solution to $A\vec{x} \approx \vec{b}$.
- When A is underdetermined, $A^+\vec{b}$ gives the least-squares solution to $A\vec{x} \approx \vec{b}$ with minimal (Euclidean) norm.

This construction from the SVD unifies solutions of the underdetermined, fully determined, and overdetermined cases of $A\vec{x} \approx \vec{b}$.

7.2.2 Decomposition into Outer Products and Low-Rank Approximations

If we expand the product $A = U\Sigma V^\top$ column by column, an equivalent formula is the following:

$$A = \sum_{i=1}^{\ell} \sigma_i \vec{u}_i \vec{v}_i^\top,$$

where $\ell \equiv \min\{m, n\}$ and $\vec{u}_i$ and $\vec{v}_i$ are the i-th columns of U and V, respectively. The sum only goes to ℓ since the remaining columns of U or V will be zeroed out by Σ.

This expression shows that any matrix can be decomposed as the sum of *outer products* of vectors:

Definition 7.2 (Outer product)**.** The *outer product* of $\vec{u} \in \mathbb{R}^m$ and $\vec{v} \in \mathbb{R}^n$ is the matrix $\vec{u} \otimes \vec{v} \equiv \vec{u}\vec{v}^\top \in \mathbb{R}^{m\times n}$.

This alternative formula for the SVD provides a new way to compute the product $A\vec{x}$:

$$A\vec{x} = \left(\sum_{i=1}^{\ell} \sigma_i \vec{u}_i \vec{v}_i^\top\right)\vec{x} = \sum_{i=1}^{\ell} \sigma_i \vec{u}_i (\vec{v}_i^\top \vec{x}) = \sum_{i=1}^{\ell} \sigma_i (\vec{v}_i \cdot \vec{x})\vec{u}_i, \text{ since } \vec{x}\cdot\vec{y} = \vec{x}^\top\vec{y}.$$

In words, applying A to $\vec{x}$ is the same as linearly combining the $\vec{u}_i$ vectors with weights $\sigma_i(\vec{v}_i \cdot \vec{x})$. This formula provides savings when the number of nonzero σ_i values is relatively small. More importantly, we can round small values of σ_i to zero, truncating this sum to *approximate* $A\vec{x}$ with fewer terms.

Similarly, from §7.2.1 we can write the pseudoinverse of A as:

$$A^+ = \sum_{\sigma_i \neq 0} \frac{\vec{v}_i \vec{u}_i^\top}{\sigma_i}.$$

With this formula, we can apply the same truncation trick to compute $A^+\vec{x}$ and can approximate $A^+\vec{x}$ by only evaluating those terms in the sum for which σ_i is relatively *small.*

In practice, we compute the singular values σ_i as square roots of eigenvalues of $A^\top A$ or $AA^\top$, and methods like power iteration can be used to reveal a partial rather than full set of eigenvalues. If we are satisfied with approximating $A^+\vec{x}$, we can compute a few of the smallest σ_i values and truncate the formula above rather than finding A^+ completely. This also avoids ever having to compute or store the full A^+ matrix and can be accurate when A has a wide range of singular values.

Returning to our original notation $A = U\Sigma V^\top$, our argument above shows that a useful approximation of A is $\tilde{A} \equiv U\tilde{\Sigma}V^\top$, where $\tilde{\Sigma}$ rounds small values of Σ to zero. The column space of $\tilde{A}$ has dimension equal to the number of nonzero values on the diagonal of $\tilde{\Sigma}$. This approximation is not an *ad hoc* estimate but rather solves a difficult optimization problem posed by the following famous theorem (stated without proof):

Theorem 7.1 (Eckart-Young, 1936)**.** Suppose $\tilde{A}$ is obtained from $A = U\Sigma V^\top$ by truncating all but the k largest singular values σ_i of A to zero. Then, $\tilde{A}$ minimizes both $\|A-\tilde{A}\|_{\text{Fro}}$ and $\|A-\tilde{A}\|_2$ subject to the constraint that the column space of $\tilde{A}$ has at most dimension k.

7.2.3 Matrix Norms

Constructing the SVD also enables us to return to our discussion of matrix norms from §4.3.1. For example, recall that the *Frobenius* norm of A is

$$\|A\|_{\text{Fro}}^2 \equiv \sum_{ij} a_{ij}^2.$$

If we write $A = U\Sigma V^\top$, we can simplify this expression:

$$\begin{aligned}
\|A\|_{\text{Fro}}^2 &= \sum_j \|A\vec{e}_j\|_2^2 \text{ since the product } A\vec{e}_j \text{ is the } j\text{-th column of } A \\
&= \sum_j \|U\Sigma V^\top \vec{e}_j\|_2^2, \text{ substituting the SVD} \\
&= \sum_j \vec{e}_j^{\,\top} V\Sigma^2 V^\top \vec{e}_j \text{ since } \|\vec{x}\|_2^2 = \vec{x}^\top\vec{x} \text{ and } U \text{ is orthogonal} \\
&= \|\Sigma V^\top\|_{\text{Fro}}^2 \text{ by reversing the steps above} \\
&= \|V\Sigma\|_{\text{Fro}}^2 \text{ since a matrix and its transpose have the same Frobenius norm} \\
&= \sum_j \|V\Sigma\vec{e}_j\|_2^2 = \sum_j \sigma_j^2 \|V\vec{e}_j\|_2^2 \text{ since } \Sigma \text{ is a diagonal matrix} \\
&= \sum_j \sigma_j^2 \text{ since } V \text{ is orthogonal.}
\end{aligned}$$

Thus, the squared Frobenius norm of $A \in \mathbb{R}^{m\times n}$ is the sum of the squares of its singular values.

This result is of theoretical interest, but it is easier to evaluate the Frobenius norm of A by summing the squares of its elements rather than finding its SVD. More interestingly, recall that the induced two-norm of A is given by

$$\|A\|_2^2 = \max\{\lambda : \text{there exists } \vec{x} \in \mathbb{R}^n \text{ with } A^\top A\vec{x} = \lambda\vec{x}\}.$$

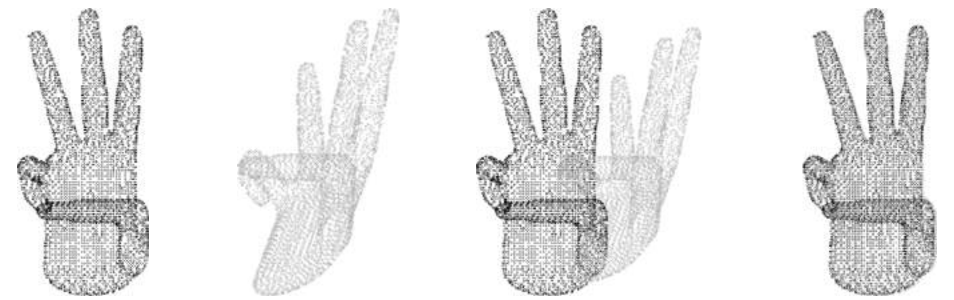


Figure 7.2 If we scan a three-dimensional object from two angles, the end result is two point clouds that are not aligned. The approach explained in §7.2.4 aligns the two clouds, serving as the first step in combining the scans. (Figure generated by S. Chung.)

In the language of the SVD, this value is the square root of the largest eigenvalue of $A^\top A$, or equivalently

$$\|A\|_2 = \max\{\sigma_i\}.$$

In other words, the induced two-norm of A can be read directly from its singular values.

Similarly, recall that the condition number of an invertible matrix A is given by cond $A = \|A\|_2\|A^{-1}\|_2$. By our derivation of A^+, the singular values of A^{-1} must be the reciprocals of the singular values of A. Combining this with the formula above for $\|A\|_2$ yields:

$$\text{cond } A = \frac{\sigma_{\max}}{\sigma_{\min}}.$$

This expression provides a new formula for evaluating the conditioning of A.

There is one caveat that prevents this formula for the condition number from being used universally. In some cases, algorithms for computing $\sigma_{\min}$ may involve solving systems $A\vec{x} = \vec{b}$, a process which in itself may suffer from poor conditioning of A. Hence, we cannot always trust values of $\sigma_{\min}$. If this is an issue, condition numbers can be bounded or approximated using various inequalities involving the singular values of A. Also, alternative iterative algorithms similar to QR iteration can be applied to computing $\sigma_{\min}$.

7.2.4 The Procrustes Problem and Point Cloud Alignment

Many techniques in computer vision involve the alignment of three-dimensional shapes. For instance, suppose we have a laser scanner that collects two point clouds of the same rigid object from different views. A typical task is to align these two point clouds into a single coordinate frame, as illustrated in Figure 7.2.

Since the object is rigid, we expect there to be some orthogonal matrix R and translation $\vec{t} \in \mathbb{R}^3$ such that rotating the first point cloud by R and then translating by $\vec{t}$ aligns the two data sets. Our job is to estimate $\vec{t}$ and R.

If the two scans overlap, the user or an automated system may mark n points that correspond between the two scans; we can store these in two matrices $X_1, X_2 \in \mathbb{R}^{3\times n}$. Then, for each column $\vec{x}_{1i}$ of X_1 and $\vec{x}_{2i}$ of X_2, we expect $R\vec{x}_{1i} + \vec{t} = \vec{x}_{2i}$. To account for error in measuring X_1 and X_2, rather than expecting exact equality, we will minimize an objective function that measures how much this relationship holds true:

$$E(R, \vec{t}) \equiv \sum_i \|R\vec{x}_{1i} + \vec{t} - \vec{x}_{2i}\|_2^2.$$

If we fix R and only consider $\vec{t}$, minimizing E becomes a least-squares problem. On the other hand, optimizing for R with $\vec{t}$ fixed is the same as minimizing $\|RX_1 - X_2^t\|_{\text{Fro}}^2$, where the columns of X_2^t are those of X_2 translated by $\vec{t}$. This second optimization is subject to the constraint that R is a 3×3 orthogonal matrix, that is, that $R^\top R = I_{3\times 3}$. It is known as the *orthogonal Procrustes problem.*

To solve this problem using the SVD, we will introduce the *trace* of a square matrix as follows:

Definition 7.3 (Trace)**.** The *trace* of $A \in \mathbb{R}^{n\times n}$ is the sum of its diagonal elements:

$$\operatorname{tr}(A) \equiv \sum_i a_{ii}.$$

In Exercise 7.2, you will check that $\|A\|_{\text{Fro}}^2 = \operatorname{tr}(A^\top A)$. Starting from this identity, E can be simplified as follows:

$$\begin{aligned}
\|RX_1 - X_2^t\|_{\text{Fro}}^2 &= \operatorname{tr}((RX_1 - X_2^t)^\top (RX_1 - X_2^t)) \\
&= \operatorname{tr}(X_1^\top X_1 - X_1^\top R^\top X_2^t - X_2^{t\top} RX_1 + X_2^{t\top} X_2) \\
&= \text{const.} - 2\operatorname{tr}(X_2^{t\top} RX_1), \\
&\qquad \text{since } \operatorname{tr}(A+B) = \operatorname{tr} A + \operatorname{tr} B \text{ and } \operatorname{tr}(A^\top) = \operatorname{tr}(A).
\end{aligned}$$

This argument shows that we wish to maximize $\operatorname{tr}(X_2^{t\top} RX_1)$ with $R^\top R = I_{3\times 3}$. From Exercise 7.2, $\operatorname{tr}(AB) = \operatorname{tr}(BA)$. Applying this identity, the objective simplifies to $\operatorname{tr}(RC)$ with $C \equiv X_1 X_2^{t\top}$. If we decompose $C = U\Sigma V^\top$ then:

$$\begin{aligned}
\operatorname{tr}(RC) &= \operatorname{tr}(RU\Sigma V^\top) \text{ by definition} \\
&= \operatorname{tr}((V^\top RU)\Sigma) \text{ since } \operatorname{tr}(AB) = \operatorname{tr}(BA) \\
&= \operatorname{tr}(\tilde{R}\Sigma) \text{ if we define } \tilde{R} = V^\top RU, \text{ which is orthogonal} \\
&= \sum_i \sigma_i \tilde{r}_{ii} \text{ since } \Sigma \text{ is diagonal.}
\end{aligned}$$

Since $\tilde{R}$ is orthogonal, its columns all have unit length. This implies that $|\tilde{r}_{ii}| \leq 1$ for all i, since otherwise the norm of column i would be too big. Since $\sigma_i \geq 0$ for all i, this argument shows that $\operatorname{tr}(RC)$ is maximized by taking $\tilde{R} = I_{3\times 3}$, which achieves that upper bound. Undoing our substitutions shows $R = V\tilde{R}U^\top = VU^\top$.

Changing notation slightly, we have derived the following fact:

Theorem 7.2 (Orthogonal Procrustes)**.** The orthogonal matrix R minimizing $\|RX - Y\|_{\text{Fro}}^2$ is given by $VU^\top$, where SVD is applied to factor $XY^\top = U\Sigma V^\top$.

Returning to the alignment problem, one typical strategy employs *alternation*:

1. Fix R and minimize E with respect to $\vec{t}$.

2. Fix the resulting $\vec{t}$ and minimize E with respect to R subject to $R^\top R = I_{3\times 3}$.

3. Return to step 1.

The energy E decreases with each step and thus converges to a local minimum. Since we never optimize $\vec{t}$ and R simultaneously, we cannot guarantee that the result is the smallest possible value of E, but in practice this method works well. Alternatively, in some cases it is possible to work out an explicit formula for $\vec{t}$, circumventing the least-squares step.

7.2.5 Principal Component Analysis (PCA)

Recall the setup from §6.1.1: We wish to find a low-dimensional approximation of a set of data points stored in the columns of a matrix $X \in \mathbb{R}^{n\times k}$, for k observations in n dimensions. Previously, we showed that if we wish to project onto a single dimension, the best possible axis is given by the dominant eigenvector of $XX^\top$. With the SVD in hand, we can consider more complicated datasets that need more than one projection axis.

Suppose that we wish to choose d vectors whose span best contains the data points in X (we considered $d = 1$ in §6.1.1); we will assume $d \leq \min\{k, n\}$. These vectors can be written in the columns of an $n \times d$ matrix C. The column space of C is preserved when we orthogonalize its columns. Rather than orthogonalizing *a posteriori*, however, we can safely restrict our search to matrices C whose columns are orthonormal, satisfying $C^\top C = I_{d\times d}$. Then, the projection of X onto the column space of C is given by $CC^\top X$.

Paralleling our earlier development, we will minimize $\|X - CC^\top X\|_{\text{Fro}}$ subject to $C^\top C = I_{d\times d}$. The objective can be simplified using trace identities:

$$\begin{aligned}
\|X - CC^\top X\|_{\text{Fro}}^2 &= \operatorname{tr}((X - CC^\top X)^\top (X - CC^\top X)) \text{ since } \|A\|_{\text{Fro}}^2 = \operatorname{tr}(A^\top A) \\
&= \operatorname{tr}(X^\top X - 2X^\top CC^\top X + X^\top CC^\top CC^\top X) \\
&= \text{const.} - \operatorname{tr}(X^\top CC^\top X) \text{ since } C^\top C = I_{d\times d} \\
&= -\|C^\top X\|_{\text{Fro}}^2 + \text{const.}
\end{aligned}$$

By this chain of equalities, an equivalent problem to the minimization posed above is to maximize $\|C^\top X\|_{\text{Fro}}^2$. For statisticians, when the rows of X have mean zero, this shows that we wish to maximize the variance of the projection $C^\top X$.

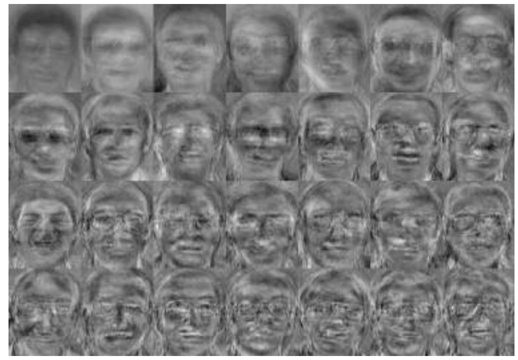
Now, introduce the SVD to factor $X = U\Sigma V^\top$. Taking $\tilde{C} \equiv U^\top C$, we are maximizing $\|C^\top U\Sigma V^\top\|_{\text{Fro}} = \|\Sigma^\top \tilde{C}\|_{\text{Fro}}$ by orthogonality of V. If the elements of $\tilde{C}$ are $\tilde{c}_{ij}$, then expanding the formula for the Frobenius norm shows

$$\|\Sigma^\top \tilde{C}\|_{\text{Fro}}^2 = \sum_i \sigma_i^2 \sum_j \tilde{c}_{ij}^2.$$

By orthogonality of the columns of $\tilde{C}$, $\sum_i \tilde{c}_{ij}^2 = 1$ for all j, and, taking into account the fact that $\tilde{C}$ may have fewer than n columns, $\sum_j \tilde{c}_{ij}^2 \leq 1$. Hence, the coefficient next to σ_i^2 is at most 1 in the sum above, and if we sort such that $\sigma_1 \geq \sigma_2 \geq \cdots$, then the maximum is achieved by taking the columns of $\tilde{C}$ to be $\vec{e}_1, \ldots, \vec{e}_d$. Undoing our change of coordinates, we see that our choice of C should be the first d columns of U.

We have shown that the SVD of X can be used to solve such a *principal component analysis* (PCA) problem. In practice, the rows of X usually are shifted to have mean zero before carrying out the SVD.

(a) Input faces



(b) Eigenfaces

(c) Projection

Figure 7.3 The "eigenface" technique [122] performs PCA on (a) a database of face images to extract (b) their most common modes of variation. For clustering, recognition, and other tasks, face images are written as (c) linear combinations of the eigenfaces, and the resulting coefficients are compared. (Figure generated by D. Hyde; images from the AT&T Database of Faces, AT&T Laboratories Cambridge.)

7.2.6 Eigenfaces*

One application of PCA in computer vision is the eigenfaces technique for face recognition, originally introduced in [122]. This popular method works by applying PCA to the images in a database of faces. Projecting new input faces onto the small PCA basis encodes a face image using just a few basis coefficients without sacrificing too much accuracy, a benefit that the method inherits from PCA.

For simplicity, suppose we have a set of k photographs of faces with similar lighting and alignment, as in Figure 7.3(a). After resizing, we can assume the photos are all of size $m \times n$, so they are representable as vectors in $\mathbb{R}^{mn}$ containing one pixel intensity per dimension. As in §7.2.5, we will store our entire database of faces in a "training matrix" $X \in \mathbb{R}^{mn\times k}$. By convention, we subtract the average face image from each column, so $X\vec{1} = \vec{0}$.

Applying PCA to X, as explained in the previous section, yields a set of "eigenface" images in the basis matrix C representing the common modes of variation between faces. One set of eigenfaces ordered by decreasing singular value is shown in Figure 7.3(b); the first few eigenfaces capture common changes in face shape, prominent features, and so on. Intuitively, PCA in this context searches for the most common distinguishing features that make a given face different from average.

The eigenface basis $C \in \mathbb{R}^{mn\times d}$ can be applied to face recognition. Suppose we take a new photo $\vec{x} \in \mathbb{R}^{mn}$ and wish to find the closest match in the database of faces. The

*Written with assistance by D. Hyde.

projection of $\vec{x}$ onto the eigenface basis is $\vec{y} \equiv C^\top \vec{x}$. The best matching face is then the closest column of $C^\top X$ to $\vec{y}$.

There are two primary advantages of eigenfaces for practical face recognition. First, usually $d \ll mn$, reducing the dimensionality of the search problem. More importantly, PCA helps separate the *relevant* modes of variation between faces from noise. Differencing the mn pixels of face images independently does not search for important facial features, while the PCA axes in C are tuned to the differences observed in the columns of X.

Many modifications, improvements, and extensions have been proposed to augment the original eigenfaces technique. For example, we can set a minimum threshold so that if the weights of a new image do not closely match any of the database weights, we report that no match was found. PCA also can be modified to be more sensitive to differences between identity rather than between lighting or pose. Even so, a rudimentary implementation is surprisingly effective. In our example, we train eigenfaces using photos of 40 subjects and then test using 40 *different* photos of the same subjects; the basic method described achieves 80% recognition accuracy.

7.3 EXERCISES

7.1 Suppose $A \in \mathbb{R}^{n\times n}$. Show that condition number of $A^\top A$ with respect to $\|\cdot\|_2$ is the square of the condition number of A

7.2 Suppose $A \in \mathbb{R}^{m\times n}$ and $B \in \mathbb{R}^{n\times m}$. Show $\|A\|_{\text{Fro}}^2 = \text{tr}(A^\top A)$ and $\text{tr}(AB) = \text{tr}(BA)$.

7.3 Provide the SVD and condition number with respect to $\|\cdot\|_2$ of the following matrices.

(a) $\begin{pmatrix} 0 & 0 & 1 \\ 0 & \sqrt{2} & 0 \\ \sqrt{3} & 0 & 0 \end{pmatrix}$

(b) $\begin{pmatrix} -5 \\ 3 \end{pmatrix}$

7.4 (Suggested by Y. Zhao.) Show that $\|A\|_2 = \|\Sigma\|_2$, where $A = U\Sigma V^T$ is the singular value decomposition of A.

7.5 Show that adding a row to a matrix cannot decrease its largest singular value.

7.6 (Suggested by Y. Zhao.) Show that the null space of a matrix $A \in \mathbb{R}^{n\times n}$ is spanned by columns of V corresponding to zero singular values, where $A = U\Sigma V^\top$ is the singular value decomposition of A.

7.7 Take $\sigma_i(A)$ to be the i-th singular value of the square matrix $A \in \mathbb{R}^{n\times n}$. Define the *nuclear norm* of A to be

$$\|A\|_* \equiv \sum_{i=1}^{n} \sigma_i(A).$$

Note: What follows is a tricky problem. Apply the mantra from this chapter: "If a linear algebra problem is hard, substitute the SVD."

(a) Show $\|A\|_* = \text{tr}(\sqrt{A^\top A})$, where trace of a matrix $\text{tr}(A)$ is the sum $\sum_i a_{ii}$ of its diagonal elements. For this problem, we will define the square root of a symmetric, positive semidefinite matrix M to be $\sqrt{M} \equiv XD^{1/2}X^\top$, where $D^{1/2}$ is the diagonal matrix containing (nonnegative) square roots of the eigenvalues of M and X contains the eigenvectors of $M = XDX^\top$.

Hint (to get started): Write $A = U\Sigma V^\top$ and argue $\Sigma^\top = \Sigma$ in this case.

(b) Show $\|A\|_* = \max_{C^\top C = I} \operatorname{tr}(AC)$.
Hint: Substitute the SVD of A and apply Exercise 7.2.

(c) Show that $\|A + B\|_* \leq \|A\|_* + \|B\|_*$.
Hint: Use Exercise 7.7b.

(d) Minimizing $\|A\vec{x} - \vec{b}\|_2^2 + \|\vec{x}\|_1$ provides an alternative to Tikhonov regularization that can yield *sparse* vectors $\vec{x}$ under certain conditions. Assuming this is the case, explain informally why minimizing $\|A - A_0\|_{\text{Fro}}^2 + \|A\|_*$ over A for a fixed $A_0 \in \mathbb{R}^{n\times n}$ might yield a *low-rank* approximation of A_0.

(e) Provide an application of solutions to the "low-rank matrix completion" problem; 7.7d provides an optimization approach to this problem.

7.8 ("Polar decomposition") In this problem we will add one more matrix factorization to our linear algebra toolbox and derive an algorithm by N. Higham for its computation [61]. The decomposition has been used in animation applications interpolating between motions of a rigid object while projecting out undesirable shearing artifacts [111].

(a) Show that any matrix $A \in \mathbb{R}^{n\times n}$ can be factored $A = WP$, where W is orthogonal and P is symmetric and positive semidefinite. This factorization is known as the polar decomposition.
Hint: Write $A = U\Sigma V^\top$ and show $V\Sigma V^\top$ is positive semidefinite.

(b) The polar decomposition of an invertible $A \in \mathbb{R}^{n\times n}$ can be computed using an iterative scheme:

$$X_0 \equiv A \qquad\qquad X_{k+1} = \frac{1}{2}(X_k + (X_k^{-1})^\top)$$

We will prove this in a few steps:

(i) Use the SVD to write $A = U\Sigma V^\top$, and define $D_k = U^\top X_k V$. Show $D_0 = \Sigma$ and $D_{k+1} = \frac{1}{2}(D_k + (D_k^{-1})^\top)$.

(ii) From (i), each D_k is diagonal. If d_{ki} is the i-th diagonal element of D_k, show

$$d_{(k+1)i} = \frac{1}{2}\left(d_{ki} + \frac{1}{d_{ki}}\right).$$

(iii) Assume $d_{ki} \to c_i$ as $k \to \infty$ (this convergence assumption requires proof!). Show $c_i = 1$.

(iv) Use 7.8(b)iii to show $X_k \to UV^\top$.

7.9 ("Derivative of SVD," [95]) In this problem, we will continue to use the notation of Exercise 4.3. Our goal is to differentiate the SVD of a matrix A with respect to changes in A. Such derivatives are used to simulate the dynamics of elastic objects; see [6] for one application.

(a) Suppose $Q(t)$ is an orthogonal matrix for all $t \in \mathbb{R}$. If we define $\Omega_Q \equiv Q^\top \partial Q$, show that Ω_Q is *antisymmetric*, that is, $\Omega_Q^\top = -\Omega_Q$. What are the diagonal elements of Ω_Q?

(b) Suppose for a matrix-valued function $A(t)$ we use SVD to decompose $A(t) = U(t)\Sigma(t)V(t)^\top$. Derive the following formula:

$$U^\top(\partial A)V = \Omega_U\Sigma + \partial\Sigma - \Sigma\Omega_V.$$

(c) Show how to compute $\partial\Sigma$ directly from ∂A and the SVD of A.

(d) Provide a method for finding Ω_U and Ω_V from ∂A and the SVD of A using a sequence of 2×2 solves. Conclude with formulas for ∂U and ∂V in terms of the Ω's.
Hint: It is sufficient to compute the elements of Ω_U and Ω_V above the diagonal.

7.10 ("Latent semantic analysis," [35]) In this problem, we explore the basics of *latent semantic analysis*, used in natural language processing to analyze collections of documents.

(a) Suppose we have a dictionary of m words and a collection of n documents. We can write an *occurrence matrix* $X \in \mathbb{R}^{m\times n}$ whose entries x_{ij} are equal to the number of times word i appears in document j. Propose interpretations of the entries of $XX^\top$ and $X^\top X$.

(b) Each document in X is represented using a point in $\mathbb{R}^m$, where m is potentially large. Suppose for efficiency and robustness to noise, we would prefer to use representations in $\mathbb{R}^k$, for some $k \ll \min\{m, n\}$. Apply Theorem 7.1 to propose a set of k vectors in $\mathbb{R}^m$ that best approximates the full space of documents with respect to the Frobenius norm.

(c) In *cross-language* applications, we might have a collection of n documents translated into two different languages, with m_1 and m_2 words, respectively. Then, we can write two occurrence matrices $X_1 \in \mathbb{R}^{m_1\times n}$ and $X_2 \in \mathbb{R}^{m_2\times n}$. Since we do not know which words in the first language correspond to which words in the second, the columns of these matrices are in correspondence but the rows are not.
One way to find similar phrases in the two languages is to find vectors $\vec{v}_1 \in \mathbb{R}^{m_1}$ and $\vec{v}_2 \in \mathbb{R}^{m_2}$ such that $X_1^\top\vec{v}_1$ and $X_2^\top\vec{v}_2$ are similar. To do so, we can solve a *canonical correlation* problem:

$$\max_{\vec{v}_1,\vec{v}_2} \frac{(X_1^\top\vec{v}_1)\cdot(X_2^\top\vec{v}_2)}{\|\vec{v}_1\|_2\|\vec{v}_2\|_2}.$$

Show how this maximization can be solved using SVD machinery.

7.11 ("Stable rank," [121]) The *stable rank* of $A \in \mathbb{R}^{n\times n}$ is defined as

$$\textsc{stable-rank}(A) \equiv \frac{\|A\|_{\text{Fro}}^2}{\|A\|_2^2}.$$

It is used in research on low-rank matrix factorization as a proxy for the rank (dimension of the column space) of A.

(a) Show that if all n columns of A are the same vector $\vec{v} \in \mathbb{R}^n\backslash\{\vec{0}\}$, then $\textsc{stable-rank}(A) = 1$.

(b) Show that when the columns of A are orthonormal, $\textsc{stable-rank}(A) = n$.

(c) More generally, show $1 \leq \textsc{stable-rank}(A) \leq n$.

(d) Show $\textsc{stable-rank}(A) \leq \textsc{rank}(A)$.