5.12 SINGULAR VALUE DECOMPOSITION

For an $m \times n$ matrix $\mathbf{A}$ of rank r, Example 5.11.2 shows how to build a URV factorization

$$\mathbf{A} = \mathbf{URV}^T = \mathbf{U}\begin{pmatrix} \mathbf{C}_{r\times r} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix}_{m\times n}\mathbf{V}^T$$

in which $\mathbf{C}$ is triangular. The purpose of this section is to prove that it's possible to do even better by showing that $\mathbf{C}$ can be made to be *diagonal*. To see how, let $\sigma_1 = \|\mathbf{A}\|_2 = \|\mathbf{C}\|_2$ (Exercise 5.6.9), and recall from the proof of (5.2.7) on p. 281 that $\|\mathbf{C}\|_2 = \|\mathbf{Cx}\|_2$ for some vector $\mathbf{x}$ such that

$$(\mathbf{C}^T\mathbf{C} - \lambda\mathbf{I})\mathbf{x} = 0, \quad \text{where} \quad \|\mathbf{x}\|_2 = 1 \text{ and } \lambda = \mathbf{x}^T\mathbf{C}^T\mathbf{Cx} = \sigma_1^2. \tag{5.12.1}$$

Set $\mathbf{y} = \mathbf{Cx}/\|\mathbf{Cx}\|_2 = \mathbf{Cx}/\sigma_1$, and let $\mathbf{R}_y = \begin{pmatrix}\mathbf{y} \,|\, \mathbf{Y}\end{pmatrix}$ and $\mathbf{R}_x = \begin{pmatrix}\mathbf{x} \,|\, \mathbf{X}\end{pmatrix}$ be elementary reflectors having $\mathbf{y}$ and $\mathbf{x}$ as their first columns, respectively—recall Example 5.6.3. Reflectors are orthogonal matrices, so $\mathbf{x}^T\mathbf{X} = \mathbf{0}$ and $\mathbf{Y}^T\mathbf{y} = \mathbf{0}$, and these together with (5.12.1) yield

$$\mathbf{y}^T\mathbf{CX} = \frac{\mathbf{x}^T\mathbf{C}^T\mathbf{CX}}{\sigma_1} = \frac{\lambda\mathbf{x}^T\mathbf{X}}{\sigma_1} = \mathbf{0} \quad \text{and} \quad \mathbf{Y}^T\mathbf{Cx} = \sigma_1\mathbf{Y}^T\mathbf{y} = \mathbf{0}.$$

Coupling these facts with $\mathbf{y}^T\mathbf{Cx} = \mathbf{y}^T(\sigma_1\mathbf{y}) = \sigma_1$ and $\mathbf{R}_y = \mathbf{R}_y^T$ produces

$$\mathbf{R}_y\mathbf{CR}_x = \begin{pmatrix}\mathbf{y}^T \\ \hline \mathbf{Y}^T\end{pmatrix}\mathbf{C}\begin{pmatrix}\mathbf{x} \,|\, \mathbf{X}\end{pmatrix} = \begin{pmatrix}\mathbf{y}^T\mathbf{Cx} & \mathbf{y}^T\mathbf{CX} \\ \mathbf{Y}^T\mathbf{Cx} & \mathbf{Y}^T\mathbf{CX}\end{pmatrix} = \begin{pmatrix}\sigma_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_2\end{pmatrix}$$

with $\sigma_1 \geq \|\mathbf{C}_2\|_2$ (because $\sigma_1 = \|\mathbf{C}\|_2 = \max\{\sigma_1, \|\mathbf{C}_2\|\}$ by (5.2.12)). Repeating the process on $\mathbf{C}_2$ yields reflectors $\mathbf{S}_y$, $\mathbf{S}_x$ such that

$$\mathbf{S}_y\mathbf{C}_2\mathbf{S}_x = \begin{pmatrix}\sigma_2 & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_3\end{pmatrix}, \quad \text{where } \sigma_2 \geq \|\mathbf{C}_3\|_2.$$

If $\mathbf{P}_2$ and $\mathbf{Q}_2$ are the orthogonal matrices

$$\mathbf{P}_2 = \begin{pmatrix}1 & 0 \\ 0 & \mathbf{S}_y\end{pmatrix}\mathbf{R}_y, \quad \mathbf{Q}_2 = \mathbf{R}_x\begin{pmatrix}1 & 0 \\ 0 & \mathbf{S}_x\end{pmatrix}, \text{ then } \mathbf{P}_2\mathbf{CQ}_2 = \begin{pmatrix}\sigma_1 & 0 & \mathbf{0} \\ 0 & \sigma_2 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{C}_3\end{pmatrix}$$

in which $\sigma_1 \geq \sigma_2 \geq \|\mathbf{C}_3\|_2$. Continuing for $r-1$ times produces orthogonal matrices $\mathbf{P}_{r-1}$ and $\mathbf{Q}_{r-1}$ such that $\mathbf{P}_{r-1}\mathbf{CQ}_{r-1} = \operatorname{diag}(\sigma_1, \sigma_2, \ldots, \sigma_r) = \mathbf{D}$, where $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r$. If $\tilde{\mathbf{U}}^T$ and $\tilde{\mathbf{V}}$ are the orthogonal matrices

$$\tilde{\mathbf{U}}^T = \begin{pmatrix}\mathbf{P}_{r-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}\end{pmatrix}\mathbf{U}^T \text{ and } \tilde{\mathbf{V}} = \mathbf{V}\begin{pmatrix}\mathbf{Q}_{r-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}\end{pmatrix}, \text{ then } \tilde{\mathbf{U}}^T\mathbf{A}\tilde{\mathbf{V}} = \begin{pmatrix}\mathbf{D} & \mathbf{0} \\ \mathbf{0} & \mathbf{0}\end{pmatrix},$$

and thus the *singular value decomposition* (SVD) is derived.[57]

[57] The SVD has been independently discovered and rediscovered several times. Those credited with the early developments include Eugenio Beltrami (1835–1899) in 1873, M. E. Camille Jordan (1838–1922) in 1875, James J. Sylvester (1814–1897) in 1889, L. Autonne in 1913, and C. Eckart and G. Young in 1936.

Singular Value Decomposition

For each $\mathbf{A} \in \Re^{m\times n}$ of rank r, there are orthogonal matrices $\mathbf{U}_{m\times m}$, $\mathbf{V}_{n\times n}$ and a diagonal matrix $\mathbf{D}_{r\times r} = \operatorname{diag}(\sigma_1, \sigma_2, \ldots, \sigma_r)$ such that

$$\mathbf{A} = \mathbf{U}\begin{pmatrix} \mathbf{D} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix}_{m\times n}\mathbf{V}^T \quad \text{with} \quad \sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0. \tag{5.12.2}$$

The σ_i's are called the nonzero ***singular values*** of $\mathbf{A}$. When $r < p = \min\{m, n\}$, $\mathbf{A}$ is said to have $p - r$ additional zero singular values. The factorization in (5.12.2) is called a ***singular value decomposition*** of $\mathbf{A}$, and the columns in $\mathbf{U}$ and $\mathbf{V}$ are called left-hand and right-hand ***singular vectors*** for $\mathbf{A}$, respectively.

While the constructive method used to derive the SVD can be used as an algorithm, more sophisticated techniques exist, and all good matrix computation packages contain numerically stable SVD implementations. However, the details of a practical SVD algorithm are too complicated to be discussed at this point.

The SVD is valid for complex matrices when $(\star)^T$ is replaced by $(\star)^*$, and it can be shown that the singular values are unique, but the singular vectors are not. In the language of Chapter 7, the σ_i^2's are the eigenvalues of $\mathbf{A}^T\mathbf{A}$, and the singular vectors are specialized sets of eigenvectors for $\mathbf{A}^T\mathbf{A}$—see the summary on p. 555. In fact, the practical algorithm for computing the SVD is an implementation of the QR iteration (p. 535) that is cleverly applied to $\mathbf{A}^T\mathbf{A}$ without ever explicitly computing $\mathbf{A}^T\mathbf{A}$.

Singular values reveal something about the geometry of linear transformations because the singular values $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n$ of a matrix $\mathbf{A}$ tell us how much distortion can occur under transformation by $\mathbf{A}$. They do so by giving us an explicit picture of how $\mathbf{A}$ distorts the unit sphere. To develop this, suppose that $\mathbf{A} \in \Re^{n\times n}$ is nonsingular (Exercise 5.12.5 treats the singular and rectangular case), and let $\mathcal{S}_2 = \{\mathbf{x} \mid \|\mathbf{x}\|_2 = 1\}$ be the unit 2-sphere in $\Re^n$. The nature of the image $\mathbf{A}(\mathcal{S}_2)$ is revealed by considering the singular value decompositions

$$\mathbf{A} = \mathbf{U}\mathbf{D}\mathbf{V}^T \quad \text{and} \quad \mathbf{A}^{-1} = \mathbf{V}\mathbf{D}^{-1}\mathbf{U}^T \quad \text{with} \quad \mathbf{D} = \operatorname{diag}(\sigma_1, \sigma_2, \ldots, \sigma_n),$$

where $\mathbf{U}$ and $\mathbf{V}$ are orthogonal matrices. For each $\mathbf{y} \in \mathbf{A}(\mathcal{S}_2)$ there is an $\mathbf{x} \in \mathcal{S}_2$ such that $\mathbf{y} = \mathbf{A}\mathbf{x}$, so, with $\mathbf{w} = \mathbf{U}^T\mathbf{y}$,

$$\begin{aligned} 1 &= \|\mathbf{x}\|_2^2 = \left\|\mathbf{A}^{-1}\mathbf{A}\mathbf{x}\right\|_2^2 = \left\|\mathbf{A}^{-1}\mathbf{y}\right\|_2^2 = \left\|\mathbf{V}\mathbf{D}^{-1}\mathbf{U}^T\mathbf{y}\right\|_2^2 = \left\|\mathbf{D}^{-1}\mathbf{U}^T\mathbf{y}\right\|_2^2 \\ &= \left\|\mathbf{D}^{-1}\mathbf{w}\right\|_2^2 = \frac{w_1^2}{\sigma_1^2} + \frac{w_2^2}{\sigma_2^2} + \cdots + \frac{w_r^2}{\sigma_r^2}. \end{aligned} \tag{5.12.3}$$

This means that $\mathbf{U}^T\mathbf{A}(\mathcal{S}_2)$ is an ellipsoid whose k^{th} semiaxis has length σ_k. Because orthogonal transformations are isometries (length preserving transformations), $\mathbf{U}^T$ can only affect the orientation of $\mathbf{A}(\mathcal{S}_2)$, so $\mathbf{A}(\mathcal{S}_2)$ is also an ellipsoid whose k^{th} semiaxis has length σ_k. Furthermore, (5.12.3) implies that the ellipsoid $\mathbf{U}^T\mathbf{A}(\mathcal{S}_2)$ is in standard position—i.e., its axes are directed along the standard basis vectors $\mathbf{e}_k$. Since $\mathbf{U}$ maps $\mathbf{U}^T\mathbf{A}(\mathcal{S}_2)$ to $\mathbf{A}(\mathcal{S}_2)$, and since $\mathbf{U}\mathbf{e}_k = \mathbf{U}_{*k}$, it follows that the axes of $\mathbf{A}(\mathcal{S}_2)$ are directed along the left-hand singular vectors defined by the columns of $\mathbf{U}$. Therefore, the k^{th} semiaxis of $\mathbf{A}(\mathcal{S}_2)$ is $\sigma_k\mathbf{U}_{*k}$. Finally, since $\mathbf{AV} = \mathbf{UD}$ implies $\mathbf{AV}_{*k} = \sigma_k\mathbf{U}_{*k}$, the right-hand singular vector $\mathbf{V}_{*k}$ is a point on $\mathcal{S}_2$ that is mapped to the k^{th} semiaxis vector on the ellipsoid $\mathbf{A}(\mathcal{S}_2)$. The picture in $\Re^3$ looks like Figure 5.12.1.

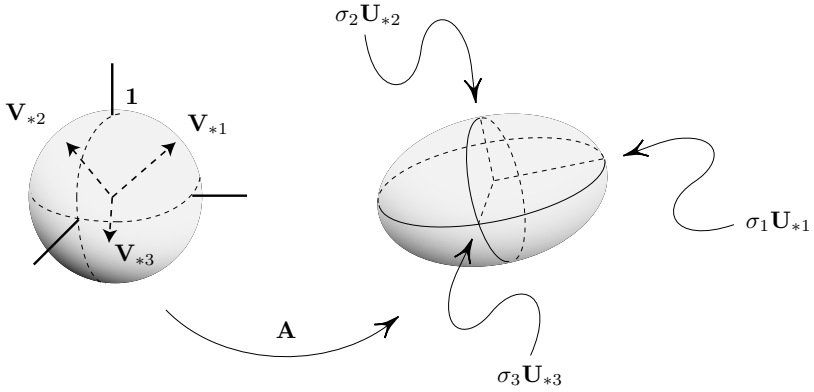

FIGURE 5.12.1

The degree of distortion of the unit sphere under transformation by $\mathbf{A}$ is therefore measured by $\kappa_2 = \sigma_1/\sigma_n$, the ratio of the largest singular value to the smallest singular value. Moreover, from the discussion of induced matrix norms (p. 280) and the unitary invariance of the 2-norm (Exercise 5.6.9),

$$\max_{\|\mathbf{x}\|_2=1} \|\mathbf{Ax}\|_2 = \|\mathbf{A}\|_2 = \left\|\mathbf{UDV}^T\right\|_2 = \|\mathbf{D}\|_2 = \sigma_1$$

and

$$\min_{\|\mathbf{x}\|_2=1} \|\mathbf{Ax}\|_2 = \frac{1}{\|\mathbf{A}^{-1}\|_2} = \frac{1}{\|\mathbf{VD}^{-1}\mathbf{U}^T\|_2} = \frac{1}{\|\mathbf{D}^{-1}\|_2} = \sigma_n.$$

In other words, longest and shortest vectors on $\mathbf{A}(\mathcal{S}_2)$ have respective lengths $\sigma_1 = \|\mathbf{A}\|_2$ and $\sigma_n = 1/\left\|\mathbf{A}^{-1}\right\|_2$ (this justifies Figure 5.2.1 on p. 281), so $\kappa_2 = \|\mathbf{A}\|_2\left\|\mathbf{A}^{-1}\right\|_2$. This is called the 2-*norm condition number* of $\mathbf{A}$. Different norms result in condition numbers with different values but with more or less the same order of magnitude as κ_2 (see Exercise 5.12.3), so the qualitative information about distortion is the same. Below is a summary.

Image of the Unit Sphere

For a nonsingular $\mathbf{A}_{n\times n}$ having singular values $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n$ and an SVD $\mathbf{A} = \mathbf{U}\mathbf{D}\mathbf{V}^T$ with $\mathbf{D} = \text{diag}\,(\sigma_1, \sigma_2, \ldots, \sigma_n)$, the image of the unit 2-sphere is an ellipsoid whose k^{th} semiaxis is given by $\sigma_k \mathbf{U}_{*k}$ (see Figure 5.12.1). Furthermore, $\mathbf{V}_{*k}$ is a point on the unit sphere such that $\mathbf{A}\mathbf{V}_{*k} = \sigma_k \mathbf{U}_{*k}$. In particular,

- $\sigma_1 = \|\mathbf{A}\mathbf{V}_{*1}\|_2 = \max_{\|\mathbf{x}\|_2=1} \|\mathbf{A}\mathbf{x}\|_2 = \|\mathbf{A}\|_2,$ (5.12.4)
- $\sigma_n = \|\mathbf{A}\mathbf{V}_{*n}\|_2 = \min_{\|\mathbf{x}\|_2=1} \|\mathbf{A}\mathbf{x}\|_2 = 1/\|\mathbf{A}^{-1}\|_2.$ (5.12.5)

The degree of distortion of the unit sphere under transformation by $\mathbf{A}$ is measured by the 2-norm ***condition number***

- $\kappa_2 = \dfrac{\sigma_1}{\sigma_n} = \|\mathbf{A}\|_2 \left\|\mathbf{A}^{-1}\right\|_2 \geq 1.$ (5.12.6)

Notice that $\kappa_2 = 1$ if and only if $\mathbf{A}$ is an orthogonal matrix.

The amount of distortion of the unit sphere under transformation by $\mathbf{A}$ determines the degree to which uncertainties in a linear system $\mathbf{A}\mathbf{x} = \mathbf{b}$ can be magnified. This is explained in the following example.

Example 5.12.1

Uncertainties in Linear Systems. Systems of linear equations $\mathbf{A}\mathbf{x} = \mathbf{b}$ arising in practical work almost always come with built-in uncertainties due to modeling errors (because assumptions are almost always necessary), data collection errors (because infinitely precise gauges don't exist), and data entry errors (because numbers like $\sqrt{2}$, π, and $2/3$ can't be entered exactly). In addition, roundoff error in floating-point computation is a prevalent source of uncertainty. In all cases it's important to estimate the degree of uncertainty in the solution of $\mathbf{A}\mathbf{x} = \mathbf{b}$. This is not difficult when $\mathbf{A}$ is known exactly and all uncertainty resides in the right-hand side. Even if this is not the case, it's sometimes possible to aggregate uncertainties and shift all of them to the right-hand side.

Problem: Let $\mathbf{A}\mathbf{x} = \mathbf{b}$ be a nonsingular system in which $\mathbf{A}$ is known exactly but $\mathbf{b}$ is subject to an uncertainty $\mathbf{e}$, and consider $\mathbf{A}\tilde{\mathbf{x}} = \mathbf{b} - \mathbf{e} = \tilde{\mathbf{b}}$. Estimate the *relative uncertainty*[58] $\|\mathbf{x} - \tilde{\mathbf{x}}\| / \|\mathbf{x}\|$ in $\mathbf{x}$ in terms of the relative uncertainty $\|\mathbf{b} - \tilde{\mathbf{b}}\| / \|\mathbf{b}\| = \|\mathbf{e}\| / \|\mathbf{b}\|$ in $\mathbf{b}$. Use any vector norm and its induced matrix norm (p. 280).

[58] Knowing the *absolute* uncertainty $\|\mathbf{x} - \tilde{\mathbf{x}}\|$ by itself may not be meaningful. For example, an absolute uncertainty of a half of an inch might be fine when measuring the distance between the earth and the moon, but it's not good in the practice of eye surgery.

Solution: Use $\|\mathbf{b}\| = \|\mathbf{Ax}\| \le \|\mathbf{A}\|\,\|\mathbf{x}\|$ with $\mathbf{x} - \tilde{\mathbf{x}} = \mathbf{A}^{-1}\mathbf{e}$ to write

$$\frac{\|\mathbf{x}-\tilde{\mathbf{x}}\|}{\|\mathbf{x}\|} = \frac{\left\|\mathbf{A}^{-1}\mathbf{e}\right\|}{\|\mathbf{x}\|} \le \frac{\|\mathbf{A}\|\left\|\mathbf{A}^{-1}\right\|\|\mathbf{e}\|}{\|\mathbf{b}\|} = \kappa\frac{\|\mathbf{e}\|}{\|\mathbf{b}\|}, \tag{5.12.7}$$

where $\kappa = \|\mathbf{A}\|\left\|\mathbf{A}^{-1}\right\|$ is a *condition number* as discussed earlier ($\kappa = \sigma_1/\sigma_n$ if the 2-norm is used). Furthermore, $\|\mathbf{e}\| = \|\mathbf{A}(\mathbf{x}-\tilde{\mathbf{x}})\| \le \|\mathbf{A}\|\,\|(\mathbf{x}-\tilde{\mathbf{x}})\|$ and $\|\mathbf{x}\| \le \left\|\mathbf{A}^{-1}\right\|\|\mathbf{b}\|$ imply

$$\frac{\|\mathbf{x}-\tilde{\mathbf{x}}\|}{\|\mathbf{x}\|} \ge \frac{\|\mathbf{e}\|}{\|\mathbf{A}\|\,\|\mathbf{x}\|} \ge \frac{\|\mathbf{e}\|}{\|\mathbf{A}\|\,\|\mathbf{A}^{-1}\|\,\|\mathbf{b}\|} = \frac{1}{\kappa}\frac{\|\mathbf{e}\|}{\|\mathbf{b}\|}.$$

This with (5.12.7) yields the following bounds on the relative uncertainty:

$$\kappa^{-1}\frac{\|\mathbf{e}\|}{\|\mathbf{b}\|} \le \frac{\|\mathbf{x}-\tilde{\mathbf{x}}\|}{\|\mathbf{x}\|} \le \kappa\frac{\|\mathbf{e}\|}{\|\mathbf{b}\|}, \quad \text{where} \quad \kappa = \|\mathbf{A}\|\left\|\mathbf{A}^{-1}\right\|. \tag{5.12.8}$$

In other words, when $\mathbf{A}$ is *well conditioned* (i.e., when κ is small—see the rule of thumb in Example 3.8.2 to get a feeling of what "small" and "large" might mean), (5.12.8) insures that small relative uncertainties in $\mathbf{b}$ cannot greatly affect the solution, but when $\mathbf{A}$ is *ill conditioned* (i.e., when κ is large), a relatively small uncertainty in $\mathbf{b}$ *might* result in a relatively large uncertainty in $\mathbf{x}$. To be more sure, the following problem needs to be addressed.

Problem: Can equality be realized in each bound in (5.12.8) for every nonsingular $\mathbf{A}$, and if so, how?

Solution: Use the 2-norm, and let $\mathbf{A} = \mathbf{UDV}^T$ be an SVD so $\mathbf{AV}_{*k} = \sigma_k\mathbf{U}_{*k}$ for each k. If $\mathbf{b}$ and $\mathbf{e}$ are directed along left-hand singular vectors associated with σ_1 and σ_n, respectively—say, $\mathbf{b} = \beta\mathbf{U}_{*1}$ and $\mathbf{e} = \epsilon\mathbf{U}_{*n}$, then

$$\mathbf{x} = \mathbf{A}^{-1}\mathbf{b} = \mathbf{A}^{-1}(\beta\mathbf{U}_{*1}) = \frac{\beta\mathbf{V}_{*1}}{\sigma_1} \quad \text{and} \quad \mathbf{x}-\tilde{\mathbf{x}} = \mathbf{A}^{-1}\mathbf{e} = \mathbf{A}^{-1}(\epsilon\mathbf{U}_{*n}) = \frac{\epsilon\mathbf{V}_{*n}}{\sigma_n},$$

so

$$\frac{\|\mathbf{x}-\tilde{\mathbf{x}}\|_2}{\|\mathbf{x}\|_2} = \left(\frac{\sigma_1}{\sigma_n}\right)\frac{|\epsilon|}{|\beta|} = \kappa_2\frac{\|\mathbf{e}\|_2}{\|\mathbf{b}\|_2} \quad \text{when } \mathbf{b} = \beta\mathbf{U}_{*1} \text{ and } \mathbf{e} = \epsilon\mathbf{U}_{*n}.$$

Thus the upper bound (the worst case) in (5.12.8) is attainable for all $\mathbf{A}$. The lower bound (the best case) is realized in the opposite situation when $\mathbf{b}$ and $\mathbf{e}$ are directed along $\mathbf{U}_{*n}$ and $\mathbf{U}_{*1}$, respectively. If $\mathbf{b} = \beta\mathbf{U}_{*n}$ and $\mathbf{e} = \epsilon\mathbf{U}_{*1}$, then the same argument yields $\mathbf{x} = \sigma_n^{-1}\beta\mathbf{V}_{*n}$ and $\mathbf{x}-\tilde{\mathbf{x}} = \sigma_1^{-1}\epsilon\mathbf{V}_{*1}$, so

$$\frac{\|\mathbf{x}-\tilde{\mathbf{x}}\|_2}{\|\mathbf{x}\|_2} = \left(\frac{\sigma_n}{\sigma_1}\right)\frac{|\epsilon|}{|\beta|} = \kappa_2^{-1}\frac{\|\mathbf{e}\|_2}{\|\mathbf{b}\|_2} \quad \text{when } \mathbf{b} = \beta\mathbf{U}_{*n} \text{ and } \mathbf{e} = \epsilon\mathbf{U}_{*1}.$$

Therefore, if $\mathbf{A}$ is well conditioned, then relatively small uncertainties in $\mathbf{b}$ can't produce relatively large uncertainties in $\mathbf{x}$. But when $\mathbf{A}$ is ill conditioned, it's possible for relatively small uncertainties in $\mathbf{b}$ to have relatively large effects on $\mathbf{x}$, and it's also possible for large uncertainties in $\mathbf{b}$ to have almost no effect on $\mathbf{x}$. Since the direction of $\mathbf{e}$ is almost always unknown, we must guard against the worst case and proceed with caution when dealing with ill-conditioned matrices.

Problem: What if there are uncertainties in both sides of $\mathbf{Ax} = \mathbf{b}$?

Solution: Use calculus to analyze the situation by considering the entries of $\mathbf{A} = \mathbf{A}(t)$ and $\mathbf{b} = \mathbf{b}(t)$ to be differentiable functions of a variable t, and compute the relative size of the derivative of $\mathbf{x} = \mathbf{x}(t)$ by differentiating $\mathbf{b} = \mathbf{Ax}$ to obtain $\mathbf{b}' = (\mathbf{Ax})' = \mathbf{A}'\mathbf{x} + \mathbf{Ax}'$ (with $\star'$ denoting $d\star/dt$), so

$$\begin{aligned}\|\mathbf{x}'\| &= \left\|\mathbf{A}^{-1}\mathbf{b}' - \mathbf{A}^{-1}\mathbf{A}'\mathbf{x}\right\| \le \left\|\mathbf{A}^{-1}\mathbf{b}'\right\| + \left\|\mathbf{A}^{-1}\mathbf{A}'\mathbf{x}\right\| \\ &\le \left\|\mathbf{A}^{-1}\right\| \|\mathbf{b}'\| + \left\|\mathbf{A}^{-1}\right\| \|\mathbf{A}'\| \|\mathbf{x}\|.\end{aligned}$$

Consequently,

$$\begin{aligned}\frac{\|\mathbf{x}'\|}{\|\mathbf{x}\|} &\le \frac{\left\|\mathbf{A}^{-1}\right\| \|\mathbf{b}'\|}{\|\mathbf{x}\|} + \left\|\mathbf{A}^{-1}\right\| \|\mathbf{A}'\| \\ &\le \|\mathbf{A}\| \left\|\mathbf{A}^{-1}\right\| \frac{\|\mathbf{b}'\|}{\|\mathbf{A}\| \|\mathbf{x}\|} + \|\mathbf{A}\| \left\|\mathbf{A}^{-1}\right\| \frac{\|\mathbf{A}'\|}{\|\mathbf{A}\|} \\ &\le \kappa \frac{\|\mathbf{b}'\|}{\|\mathbf{b}\|} + \kappa \frac{\|\mathbf{A}'\|}{\|\mathbf{A}\|} = \kappa \left(\frac{\|\mathbf{b}'\|}{\|\mathbf{b}\|} + \frac{\|\mathbf{A}'\|}{\|\mathbf{A}\|} \right).\end{aligned}$$

In other words, the relative sensitivity of the solution is the sum of the relative sensitivities of $\mathbf{A}$ and $\mathbf{b}$ magnified by $\kappa = \|\mathbf{A}\| \left\|\mathbf{A}^{-1}\right\|$. A discrete analog of the above inequality is developed in Exercise 5.12.12.

Conclusion: In all cases, the credibility of the solution to $\mathbf{Ax} = \mathbf{b}$ in the face of uncertainties must be gauged in relation to the condition of $\mathbf{A}$.

As the next example shows, the condition number is pivotal also in determining whether or not the residual $\mathbf{r} = \mathbf{b} - \mathbf{A}\tilde{\mathbf{x}}$ is a reliable indicator of the accuracy of an approximate solution $\tilde{\mathbf{x}}$.

Example 5.12.2

Checking an Answer. Suppose that $\tilde{\mathbf{x}}$ is a computed (or otherwise approximate) solution for a nonsingular system $\mathbf{Ax} = \mathbf{b}$, and suppose the accuracy of $\tilde{\mathbf{x}}$ is "checked" by computing the ***residual*** $\mathbf{r} = \mathbf{b} - \mathbf{A}\tilde{\mathbf{x}}$. If $\mathbf{r} = \mathbf{0}$, *exactly,* then $\tilde{\mathbf{x}}$ must be the exact solution. But if $\mathbf{r}$ is not exactly zero—say, $\|\mathbf{r}\|_2$ is zero to t significant digits—are we guaranteed that $\tilde{\mathbf{x}}$ is accurate to roughly t significant figures? This question was briefly examined in Example 1.6.3, but it's worth another look.

Problem: To what extent does the size of the residual reflect the accuracy of an approximate solution?

Solution: Without realizing it, we answered this question in Example 5.12.1. To bound the accuracy of $\tilde{\mathbf{x}}$ relative to the exact solution $\mathbf{x}$, write $\mathbf{r} = \mathbf{b} - \mathbf{A}\tilde{\mathbf{x}}$ as $\mathbf{A}\tilde{\mathbf{x}} = \mathbf{b} - \mathbf{r}$, and apply (5.12.8) with $\mathbf{e} = \mathbf{r}$ to obtain

$$\kappa^{-1}\frac{\|\mathbf{r}\|_2}{\|\mathbf{b}\|_2} \le \frac{\|\mathbf{x}-\tilde{\mathbf{x}}\|}{\|\mathbf{x}\|} \le \kappa\frac{\|\mathbf{r}\|_2}{\|\mathbf{b}\|_2}, \quad \text{where} \quad \kappa = \|\mathbf{A}\|_2\left\|\mathbf{A}^{-1}\right\|_2. \tag{5.12.9}$$

Therefore, for a well-conditioned $\mathbf{A}$, the residual $\mathbf{r}$ is relatively small if and only if $\tilde{\mathbf{x}}$ is relatively accurate. However, as demonstrated in Example 5.12.1, equality on either side of (5.12.9) is possible, so, when $\mathbf{A}$ is ill conditioned, a very inaccurate approximation $\tilde{\mathbf{x}}$ can produce a small residual $\mathbf{r}$, and a very accurate approximation can produce a large residual.

Conclusion: Residuals are reliable indicators of accuracy only when $\mathbf{A}$ is well conditioned—if $\mathbf{A}$ is ill conditioned, residuals are nearly meaningless.

In addition to measuring the distortion of the unit sphere and gauging the sensitivity of linear systems, singular values provide a measure of how close $\mathbf{A}$ is to a matrix of lower rank.

Distance to Lower-Rank Matrices

If $\sigma_1 \ge \sigma_2 \ge \cdots \ge \sigma_r$ are the nonzero singular values of $\mathbf{A}_{m\times n}$, then for each $k < r$, the distance from $\mathbf{A}$ to the closest matrix of rank k is

$$\sigma_{k+1} = \min_{rank(\mathbf{B})=k} \|\mathbf{A}-\mathbf{B}\|_2. \tag{5.12.10}$$

Proof. Suppose $rank\,(\mathbf{B}_{m\times n}) = k$, and let $\mathbf{A} = \mathbf{U}\begin{pmatrix}\mathbf{D} & \mathbf{0}\\ \mathbf{0} & \mathbf{0}\end{pmatrix}\mathbf{V}^T$ be an SVD for $\mathbf{A}$ with $\mathbf{D} = \text{diag}\,(\sigma_1, \sigma_2, \ldots, \sigma_r)$. Define $\mathbf{S} = \text{diag}\,(\sigma_1, \ldots, \sigma_{k+1})$, and partition $\mathbf{V} = \begin{pmatrix}\mathbf{F}_{n\times k+1} \,|\, \mathbf{G}\end{pmatrix}$. Since $rank\,(\mathbf{BF}) \le rank\,(\mathbf{B}) = k$ (by (4.5.2)), $\dim N\,(\mathbf{BF}) = k+1-rank\,(\mathbf{BF}) \ge 1$, so there is an $\mathbf{x} \in N\,(\mathbf{BF})$ with $\|\mathbf{x}\|_2 = 1$. Consequently, $\mathbf{BFx} = \mathbf{0}$ and

$$\mathbf{AFx} = \mathbf{U}\begin{pmatrix}\mathbf{D} & \mathbf{0}\\ \mathbf{0} & \mathbf{0}\end{pmatrix}\mathbf{V}^T\mathbf{Fx} = \mathbf{U}\begin{pmatrix}\mathbf{S} & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \star & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \mathbf{0}\end{pmatrix}\begin{pmatrix}\mathbf{x}\\ \mathbf{0}\\ \mathbf{0}\end{pmatrix} = \mathbf{U}\begin{pmatrix}\mathbf{Sx}\\ \mathbf{0}\\ \mathbf{0}\end{pmatrix}.$$

Since $\|\mathbf{A}-\mathbf{B}\|_2 = \max_{\|\mathbf{y}\|_2=1}\|(\mathbf{A}-\mathbf{B})\mathbf{y}\|_2$, and since $\|\mathbf{Fx}\|_2 = \|\mathbf{x}\|_2 = 1$ (recall (5.2.4), p. 280, and (5.2.13), p. 283),

$$\|\mathbf{A}-\mathbf{B}\|_2^2 \ge \|(\mathbf{A}-\mathbf{B})\mathbf{Fx}\|_2^2 = \|\mathbf{Sx}\|_2^2 = \sum_{i=1}^{k+1}\sigma_i^2x_i^2 \ge \sigma_{k+1}^2\sum_{i=1}^{k+1}x_i^2 = \sigma_{k+1}^2.$$

Equality holds for $\mathbf{B}_k = \mathbf{U}\begin{pmatrix}\mathbf{D}_k & \mathbf{0}\\ \mathbf{0} & \mathbf{0}\end{pmatrix}\mathbf{V}^T$ with $\mathbf{D}_k = \text{diag}\,(\sigma_1, \ldots, \sigma_k)$, and thus (5.12.10) is proven. ∎

Example 5.12.3

Filtering Noisy Data. The SVD can be a useful tool in applications involving the need to sort through noisy data and lift out relevant information. Suppose that $\mathbf{A}_{m\times n}$ is a matrix containing data that are contaminated with a certain level of noise—e.g., the entries $\mathbf{A}$ might be digital samples of a noisy video or audio signal such as that in Example 5.8.3 (p. 359). The SVD resolves the data in $\mathbf{A}$ into r mutually orthogonal components by writing

$$\mathbf{A} = \mathbf{U}\begin{pmatrix} \mathbf{D}_{r\times r} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix}\mathbf{V}^T = \sum_{i=1}^{r}\sigma_i\mathbf{u}_i\mathbf{v}_i^T = \sum_{i=1}^{r}\sigma_i\mathbf{Z}_i, \tag{5.12.11}$$

where $\mathbf{Z}_i = \mathbf{u}_i\mathbf{v}_i^T$ and $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0$. The matrices $\{\mathbf{Z}_1, \mathbf{Z}_2, \ldots, \mathbf{Z}_r\}$ constitute an orthonormal set because

$$\langle \mathbf{Z}_i | \mathbf{Z}_j \rangle = trace\left(\mathbf{Z}_i^{\mathbf{T}}\mathbf{Z}_j\right) = \begin{cases} 0 & \text{if } i \neq j, \\ 1 & \text{if } i = j. \end{cases}$$

In other words, the SVD (5.12.11) can be regarded as a Fourier expansion as described on p. 299 and, consequently, $\sigma_i = \langle \mathbf{Z}_i | \mathbf{A} \rangle$ can be interpreted as the proportion of $\mathbf{A}$ lying in the "direction" of $\mathbf{Z}_i$. In many applications the noise contamination in $\mathbf{A}$ is random (or nondirectional) in the sense that the noise is distributed more or less uniformly across the $\mathbf{Z}_i$'s. That is, there is about as much noise in the "direction" of one $\mathbf{Z}_i$ as there is in the "direction" of any other. Consequently, we expect each term $\sigma_i\mathbf{Z}_i$ to contain approximately the same level of noise. This means that if $\text{SNR}(\sigma_i\mathbf{Z}_i)$ denotes the *signal-to-noise ratio* in $\sigma_i\mathbf{Z}_i$, then

$$\text{SNR}(\sigma_1\mathbf{Z}_1) \geq \text{SNR}(\sigma_2\mathbf{Z}_2) \geq \cdots \geq \text{SNR}(\sigma_r\mathbf{Z}_r),$$

more or less. If some of the singular values, say, $\sigma_{k+1}, \ldots, \sigma_r$, are small relative to (total noise)$/r$, then the terms $\sigma_{k+1}\mathbf{Z}_{k+1}, \ldots, \sigma_r\mathbf{Z}_r$ have small signal-to-noise ratios. Therefore, if we delete these terms from (5.12.11), then we lose a small part of the total signal, but we remove a disproportionately large component of the total noise in $\mathbf{A}$. This explains why a *truncated* SVD $\mathbf{A}_k = \sum_{i=1}^{k}\sigma_i\mathbf{Z}_i$ can, in many instances, filter out some of the noise without losing significant information about the signal in $\mathbf{A}$. Determining the best value of k often requires empirical techniques that vary from application to application, but looking for obvious gaps between large and small singular values is usually a good place to start. The next example presents an interesting application of this idea to building an Internet search engine.

Example 5.12.4

Search Engines. The filtering idea presented in Example 5.12.3 is widely used, but a particularly novel application is the method of *latent semantic indexing* used in the areas of information retrieval and text mining. You can think of this in terms of building an Internet search engine. Start with a dictionary of terms $T_1, T_2, \ldots, T_m$. Terms are usually single words, but sometimes a term may contain more that one word such as "landing gear." It's up to you to decide how extensive your dictionary should be, but even if you use the entire English language, you probably won't be using more than a few hundred-thousand terms, and this is within the capacity of existing computer technology. Each document (or web page) D_j of interest is scanned for key terms (this is called *indexing* the document), and an associated *document vector* $\mathbf{d}_j = (\text{freq}_{1j}, \text{freq}_{2j}, \ldots, \text{freq}_{mj})^T$ is created in which

$$\text{freq}_{ij} = \text{number of times term } T_i \text{ occurs in document } D_j.$$

(More sophisticated search engines use weighted frequency strategies.) After a collection of documents $D_1, D_2, \ldots, D_n$ has been indexed, the associated document vectors $\mathbf{d}_j$ are placed as columns in a *term-by-document matrix*

$$\mathbf{A}_{m\times n} = \left(\mathbf{d}_1 \,|\, \mathbf{d}_2 \,|\, \cdots \,|\, \mathbf{d}_n\right) = \begin{array}{c} \\ T_1 \\ T_2 \\ \vdots \\ T_m \end{array} \begin{array}{c} \begin{array}{cccc} D_1 & D_2 & \cdots & D_n \end{array} \\ \begin{pmatrix} \text{freq}_{11} & \text{freq}_{12} & \cdots & \text{freq}_{1n} \\ \text{freq}_{21} & \text{freq}_{22} & \cdots & \text{freq}_{2n} \\ \vdots & \vdots & & \vdots \\ \text{freq}_{m1} & \text{freq}_{m2} & \cdots & \text{freq}_{mn} \end{pmatrix} \end{array}.$$

Naturally, most entries in each document vector $\mathbf{d}_j$ will be zero, so $\mathbf{A}$ is a sparse matrix—this is good because it means that sparse matrix technology can be applied. When a query composed of a few terms is submitted to the search engine, a *query vector* $\mathbf{q}^T = (q_1, q_2, \ldots, q_n)$ is formed in which

$$q_i = \begin{cases} 1 & \text{if term } T_i \text{ appears in the query,} \\ 0 & \text{otherwise.} \end{cases}$$

(The q_i's might also be weighted.) To measure how well a query $\mathbf{q}$ matches a document D_j, we check how close $\mathbf{q}$ is to $\mathbf{d}_j$ by computing the magnitude of

$$\cos\theta_j = \frac{\mathbf{q}^T\mathbf{d}_j}{\|\mathbf{q}\|_2\,\|\mathbf{d}_j\|_2} = \frac{\mathbf{q}^T\mathbf{A}\mathbf{e}_j}{\|\mathbf{q}\|_2\,\|\mathbf{A}\mathbf{e}_j\|_2}. \tag{5.12.12}$$

If $|\cos\theta_j| \geq \tau$ for some threshold tolerance τ, then document D_j is considered relevant and is returned to the user. Selecting τ is part art and part science that's based on experimentation and desired performance criteria. If the columns of $\mathbf{A}$ along with $\mathbf{q}$ are initially normalized to have unit length, then

$|\mathbf{q}^T\mathbf{A}| = \big(|\cos\theta_1|, |\cos\theta_2|, \ldots, |\cos\theta_n|\big)$ provides the information that allows the search engine to rank the relevance of each document relative to the query. However, due to things like variation and ambiguity in the use of vocabulary, presentation style, and even the indexing process, there is a lot of "noise" in $\mathbf{A}$, so the results in $|\mathbf{q}^T\mathbf{A}|$ are nowhere near being an exact measure of how well query $\mathbf{q}$ matches the various documents. To filter out some of this noise, the techniques of Example 5.12.3 are employed. An SVD $\mathbf{A} = \sum_{i=1}^{r} \sigma_i \mathbf{u}_i \mathbf{v}_i^T$ is judiciously truncated, and

$$\mathbf{A}_k = \mathbf{U}_k\mathbf{D}_k\mathbf{V}_k^T = \big(\mathbf{u}_1 \,|\, \cdots \,|\, \mathbf{u}_k\big) \begin{pmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_k \end{pmatrix} \begin{pmatrix} \mathbf{v}_1^T \\ \vdots \\ \mathbf{v}_k^T \end{pmatrix} = \sum_{i=1}^{k} \sigma_i \mathbf{u}_i \mathbf{v}_i^T$$

is used in place of $\mathbf{A}$ in (5.12.12). In other words, instead of using $\cos\theta_j$, query $\mathbf{q}$ is compared with document D_j by using the magnitude of

$$\cos\phi_j = \frac{\mathbf{q}^T\mathbf{A}_k\mathbf{e}_j}{\|\mathbf{q}\|_2 \, \|\mathbf{A}_k\mathbf{e}_j\|_2}.$$

To make this more suitable for computation, set $\mathbf{S}_k = \mathbf{D}_k\mathbf{V}_k^T = \big(\mathbf{s}_1 \,|\, \mathbf{s}_2 \,|\, \cdots \,|\, \mathbf{s}_k\big)$, and use

$$\|\mathbf{A}_k\mathbf{e}_j\|_2 = \big\|\mathbf{U}_k\mathbf{D}_k\mathbf{V}_k^T\mathbf{e}_j\big\|_2 = \|\mathbf{U}_k\mathbf{s}_j\|_2 = \|\mathbf{s}_j\|_2$$

to write

$$\cos\phi_j = \frac{\mathbf{q}^T\mathbf{U}_k\mathbf{s}_j}{\|\mathbf{q}\|_2 \, \|\mathbf{s}_j\|_2}. \tag{5.12.13}$$

The vectors in $\mathbf{U}_k$ and $\mathbf{S}_k$ only need to be computed once (and they can be determined without computing the entire SVD), so (5.12.13) requires very little computation to process each new query. Furthermore, we can be generous in the number of SVD components that are dropped because variation in the use of vocabulary and the ambiguity of many words produces significant noise in $\mathbf{A}$. Coupling this with the fact that numerical accuracy is not an important issue (knowing a cosine to two or three significant digits is sufficient) means that we are more than happy to replace the SVD of $\mathbf{A}$ by a low-rank truncation $\mathbf{A}_k$, where k is *significantly* less than r.

Alternate Query Matching Strategy. An alternate way to measuring how close a given query $\mathbf{q}$ is to a document vector $\mathbf{d}_j$ is to replace the query vector $\mathbf{q}$ in (5.12.12) by the *projected query* $\widetilde{\mathbf{q}} = \mathbf{P}_{R(\mathbf{A})}\mathbf{q}$, where $\mathbf{P}_{R(\mathbf{A})} = \mathbf{U}_r\mathbf{U}_r^T$ is the orthogonal projector onto $R(\mathbf{A})$ along $R(\mathbf{A})^{\perp}$ (Exercise 5.12.15) to produce

$$\cos\widetilde{\theta}_j = \frac{\widetilde{\mathbf{q}}^T\mathbf{A}\mathbf{e}_j}{\|\widetilde{\mathbf{q}}\|_2 \, \|\mathbf{A}\mathbf{e}_j\|_2}. \tag{5.12.14}$$

It's proven on p. 435 that $\widetilde{\mathbf{q}} = \mathbf{P}_{R(\mathbf{A})}\mathbf{q}$ is the vector in $R(\mathbf{A})$ (the *document space*) that is closest to $\mathbf{q}$, so using $\widetilde{\mathbf{q}}$ in place of $\mathbf{q}$ has the effect of using the best approximation to $\mathbf{q}$ that is a linear combination of the document vectors $\mathbf{d}_i$. Since $\widetilde{\mathbf{q}}^T\mathbf{A} = \mathbf{q}^T\mathbf{A}$ and $\|\widetilde{\mathbf{q}}\|_2 \leq \|\mathbf{q}\|_2$, it follows that $\cos\widetilde{\theta}_j \geq \cos\theta_j$, so more documents are deemed relevant when the projected query is used. Just as in the unprojected query matching strategy, the noise is filtered out by replacing $\mathbf{A}$ in (5.12.14) with a truncated SVD $\mathbf{A}_k = \sum_{i=1}^{k}\sigma_i\mathbf{u}_i\mathbf{v}_i^T$. The result is

$$\cos\widetilde{\phi}_j = \frac{\mathbf{q}^T\mathbf{U}_k\mathbf{s}_j}{\left\|\mathbf{U}_k^T\mathbf{q}\right\|_2\|\mathbf{s}_j\|_2}$$

and, just as in (5.12.13), $\cos\widetilde{\phi}_j$ is easily and quickly computed for each new query $\mathbf{q}$ because $\mathbf{U}_k$ and $\mathbf{s}_j$ need only be computed once.

The next example shows why singular values are the primary mechanism for numerically determining the rank of a matrix.

Example 5.12.5

Perturbations and Numerical Rank. For $\mathbf{A} \in \Re^{m\times n}$ with $p = \min\{m,n\}$, let $\{\sigma_1,\sigma_2,\ldots,\sigma_p\}$ and $\{\beta_1,\beta_2,\ldots,\beta_p\}$ be all singular values (nonzero as well as any zero ones) for $\mathbf{A}$ and $\mathbf{A}+\mathbf{E}$, respectively.

Problem: Prove that

$$|\sigma_k - \beta_k| \leq \|\mathbf{E}\|_2 \quad \text{for each} \quad k = 1,2,\ldots,p. \tag{5.12.15}$$

Solution: If the SVD for $\mathbf{A}$ given in (5.12.2) is written in the form

$$\mathbf{A} = \sum_{i=1}^{p}\sigma_i\mathbf{u}_i\mathbf{v}_i^T, \quad \text{and if we set} \quad \mathbf{A}_{k-1} = \sum_{i=1}^{k-1}\sigma_i\mathbf{u}_i\mathbf{v}_i^T,$$

then

$$\begin{aligned}\sigma_k &= \|\mathbf{A}-\mathbf{A}_{k-1}\|_2 = \|\mathbf{A}+\mathbf{E}-\mathbf{A}_{k-1}-\mathbf{E}\|_2\\ &\geq \|\mathbf{A}+\mathbf{E}-\mathbf{A}_{k-1}\|_2 - \|\mathbf{E}\|_2 \quad \text{(recall (5.1.6) on p. 273)}\\ &\geq \beta_k - \|\mathbf{E}\|_2 \quad \text{by (5.12.10).}\end{aligned}$$

Couple this with the observation that

$$\begin{aligned}\sigma_k &= \min_{rank(\mathbf{B})=k-1}\|\mathbf{A}-\mathbf{B}\|_2 = \min_{rank(\mathbf{B})=k-1}\|\mathbf{A}+\mathbf{E}-\mathbf{B}-\mathbf{E}\|_2\\ &\leq \min_{rank(\mathbf{B})=k-1}\|\mathbf{A}+\mathbf{E}-\mathbf{B}\|_2 + \|\mathbf{E}\|_2 = \beta_k + \|\mathbf{E}\|_2\end{aligned}$$

to conclude that $|\sigma_k - \beta_k| \leq \|\mathbf{E}\|_2$.

Problem: Explain why this means that computing the singular values of $\mathbf{A}$ with any stable algorithm (one that returns the exact singular values β_k of a nearby matrix $\mathbf{A}+\mathbf{E}$) is a good way to compute $rank\,(\mathbf{A})$.

Solution: If $rank\,(\mathbf{A}) = r$, then $p-r$ of the σ_k's are exactly zero, so the perturbation result (5.12.15) guarantees that $p-r$ of the computed β_k's cannot be larger than $\|\mathbf{E}\|_2$. So if

$$\beta_1 \geq \cdots \geq \beta_{\tilde{r}} > \|\mathbf{E}\|_2 \geq \beta_{\tilde{r}+1} \geq \cdots \geq \beta_p,$$

then it's reasonable to consider $\tilde{r}$ to be the ***numerical rank*** of $\mathbf{A}$. For most algorithms, $\|\mathbf{E}\|_2$ is not known exactly, but adequate estimates of $\|\mathbf{E}\|_2$ often can be derived. Considerable effort has gone into the development of stable algorithms for computing singular values, but such algorithms are too involved to discuss here—consult an advanced book on matrix computations. Generally speaking, good SVD algorithms have $\|\mathbf{E}\|_2 \approx 5\times 10^{-t}\|\mathbf{A}\|_2$ when t-digit floating-point arithmetic is used.

Just as the range-nullspace decomposition was used in Example 5.10.5 to define the Drazin inverse of a square matrix, a URV factorization or an SVD can be used to define a generalized inverse for rectangular matrices. For a URV factorization

$$\mathbf{A}_{m\times n} = \mathbf{U}\begin{pmatrix}\mathbf{C} & \mathbf{0}\\ \mathbf{0} & \mathbf{0}\end{pmatrix}\mathbf{V}^T_{m\times n}, \quad \text{we define} \quad \mathbf{A}^{\dagger}_{n\times m} = \mathbf{V}\begin{pmatrix}\mathbf{C}^{-1} & \mathbf{0}\\ \mathbf{0} & \mathbf{0}\end{pmatrix}\mathbf{U}^T_{n\times m}$$

to be the ***Moore–Penrose inverse*** (or the ***pseudoinverse***) of $\mathbf{A}$. (Replace $(\star)^T$ by $(\star)^*$ when $\mathbf{A}\in\mathcal{C}^{m\times n}$.) Although the URV factors are not uniquely defined by $\mathbf{A}$, it can be proven that $\mathbf{A}^{\dagger}$ is unique by arguing that $\mathbf{A}^{\dagger}$ is the unique solution to the four Penrose equations

$$\mathbf{A}\mathbf{A}^{\dagger}\mathbf{A} = \mathbf{A}, \qquad \mathbf{A}^{\dagger}\mathbf{A}\mathbf{A}^{\dagger} = \mathbf{A}^{\dagger},$$

$$\left(\mathbf{A}\mathbf{A}^{\dagger}\right)^T = \mathbf{A}\mathbf{A}^{\dagger}, \qquad \left(\mathbf{A}^{\dagger}\mathbf{A}\right)^T = \mathbf{A}^{\dagger}\mathbf{A},$$

so $\mathbf{A}^{\dagger}$ is the same matrix defined in Exercise 4.5.20. Since it doesn't matter which URV factorization is used, we can use the SVD (5.12.2), in which case $\mathbf{C} = \mathbf{D} = \text{diag}\,(\sigma_1,\ldots,\sigma_r)$. Some "inverselike" properties that relate $\mathbf{A}^{\dagger}$ to solutions and least squares solutions for linear systems are given in the following summary. Other useful properties appear in the exercises.

Moore–Penrose Pseudoinverse

- In terms of URV factors, the Moore–Penrose pseudoinverse of

$$\mathbf{A}_{m\times n} = \mathbf{U}\begin{pmatrix}\mathbf{C}_{r\times r} & \mathbf{0}\\ \mathbf{0} & \mathbf{0}\end{pmatrix}\mathbf{V}^T \text{ is } \mathbf{A}^{\dagger}_{n\times m} = \mathbf{V}\begin{pmatrix}\mathbf{C}^{-1} & \mathbf{0}\\ \mathbf{0} & \mathbf{0}\end{pmatrix}\mathbf{U}^T. \quad (5.12.16)$$

- When $\mathbf{Ax}=\mathbf{b}$ is consistent, $\mathbf{x}=\mathbf{A}^{\dagger}\mathbf{b}$ is the solution of minimal euclidean norm. (5.12.17)
- When $\mathbf{Ax}=\mathbf{b}$ is inconsistent, $\mathbf{x}=\mathbf{A}^{\dagger}\mathbf{b}$ is the least squares solution of minimal euclidean norm. (5.12.18)
- When an SVD is used, $\mathbf{C}=\mathbf{D}=\operatorname{diag}(\sigma_1,\ldots,\sigma_r)$, so

$$\mathbf{A}^{\dagger} = \mathbf{V}\begin{pmatrix}\mathbf{D}^{-1} & \mathbf{0}\\ \mathbf{0} & \mathbf{0}\end{pmatrix}\mathbf{U}^T = \sum_{i=1}^{r}\frac{\mathbf{v}_i\mathbf{u}_i^T}{\sigma_i} \quad \text{and} \quad \mathbf{A}^{\dagger}\mathbf{b} = \sum_{i=1}^{r}\frac{(\mathbf{u}_i^T\mathbf{b})}{\sigma_i}\mathbf{v}_i.$$

Proof. To prove (5.12.17), suppose $\mathbf{Ax}_0=\mathbf{b}$, and replace $\mathbf{A}$ by $\mathbf{AA}^{\dagger}\mathbf{A}$ to write $\mathbf{b}=\mathbf{Ax}_0=\mathbf{AA}^{\dagger}\mathbf{Ax}_0=\mathbf{AA}^{\dagger}\mathbf{b}$. Thus $\mathbf{A}^{\dagger}\mathbf{b}$ solves $\mathbf{Ax}=\mathbf{b}$ when it is consistent. To see that $\mathbf{A}^{\dagger}\mathbf{b}$ is the solution of minimal norm, observe that the general solution is $\mathbf{A}^{\dagger}\mathbf{b}+N(\mathbf{A})$ (a particular solution plus the general solution of the homogeneous equation), so every solution has the form $\mathbf{z}=\mathbf{A}^{\dagger}\mathbf{b}+\mathbf{n}$, where $\mathbf{n}\in N(\mathbf{A})$. It's not difficult to see that $\mathbf{A}^{\dagger}\mathbf{b}\in R\left(\mathbf{A}^{\dagger}\right)=R\left(\mathbf{A}^T\right)$ (Exercise 5.12.16), so $\mathbf{A}^{\dagger}\mathbf{b}\perp\mathbf{n}$. Therefore, by the Pythagorean theorem (Exercise 5.4.14),

$$\|\mathbf{z}\|_2^2 = \left\|\mathbf{A}^{\dagger}\mathbf{b}+\mathbf{n}\right\|_2^2 = \left\|\mathbf{A}^{\dagger}\mathbf{b}\right\|_2^2 + \|\mathbf{n}\|_2^2 \geq \left\|\mathbf{A}^{\dagger}\mathbf{b}\right\|_2^2.$$

Equality is possible if and only if $\mathbf{n}=\mathbf{0}$, so $\mathbf{A}^{\dagger}\mathbf{b}$ is the *unique* minimum norm solution. When $\mathbf{Ax}=\mathbf{b}$ is inconsistent, the least squares solutions are the solutions of the normal equations $\mathbf{A}^T\mathbf{Ax}=\mathbf{A}^T\mathbf{b}$, and it's straightforward to verify that $\mathbf{A}^{\dagger}\mathbf{b}$ is one such solution (Exercise 5.12.16(c)). To prove that $\mathbf{A}^{\dagger}\mathbf{b}$ is the least squares solution of minimal norm, apply the same argument used in the consistent case to the normal equations. ∎

Caution! Generalized inverses are useful in formulating theoretical statements such as those above, but, just as in the case of the ordinary inverse, generalized inverses are not practical computational tools. In addition to being computationally inefficient, serious numerical problems result from the fact that $\mathbf{A}^{\dagger}$ need

not be a continuous function of the entries of $\mathbf{A}$. For example,

$$\mathbf{A}(x) = \begin{pmatrix} 1 & 0 \\ 0 & x \end{pmatrix} \implies \mathbf{A}^{\dagger}(x) = \begin{cases} \begin{pmatrix} 1 & 0 \\ 0 & 1/x \end{pmatrix} & \text{for } x \neq 0, \\ \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} & \text{for } x = 0. \end{cases}$$

Not only is $\mathbf{A}^{\dagger}(x)$ discontinuous in the sense that $\lim_{x\to 0} \mathbf{A}^{\dagger}(x) \neq \mathbf{A}^{\dagger}(0)$, but it is discontinuous in the worst way because as $\mathbf{A}(x)$ comes closer to $\mathbf{A}(0)$ the matrix $\mathbf{A}^{\dagger}(x)$ moves farther away from $\mathbf{A}^{\dagger}(0)$. This type of behavior translates into insurmountable computational difficulties because small errors due to round-off (or anything else) can produce enormous errors in the computed $\mathbf{A}^{\dagger}$, and as errors in $\mathbf{A}$ become smaller the resulting errors in $\mathbf{A}^{\dagger}$ can become greater. This diabolical fact is also true for the Drazin inverse (p. 399). The inherent numerical problems coupled with the fact that it's extremely rare for an application to require explicit knowledge of the entries of $\mathbf{A}^{\dagger}$ or $\mathbf{A}^{D}$ constrains them to being theoretical or notational tools. But don't underestimate this role—go back and read Laplace's statement quoted in the footnote on p. 81.

Example 5.12.6

Another way to view the URV or SVD factorizations in relation to the Moore–Penrose inverse is to consider $\mathbf{A}_{/R(\mathbf{A}^T)}$ and $\mathbf{A}^{\dagger}_{/R(\mathbf{A})}$, the restrictions of $\mathbf{A}$ and $\mathbf{A}^{\dagger}$ to $R\left(\mathbf{A}^T\right)$ and $R(\mathbf{A})$, respectively. Begin by making the straightforward observations that $R\left(\mathbf{A}^{\dagger}\right) = R\left(\mathbf{A}^T\right)$ and $N\left(\mathbf{A}^{\dagger}\right) = N\left(\mathbf{A}^T\right)$ (Exercise 5.12.16). Since $\Re^n = R\left(\mathbf{A}^T\right) \oplus N(\mathbf{A})$ and $\Re^m = R(\mathbf{A}) \oplus N\left(\mathbf{A}^T\right)$, it follows that $R(\mathbf{A}) = \mathbf{A}(\Re^n) = \mathbf{A}(R\left(\mathbf{A}^T\right))$ and $R\left(\mathbf{A}^T\right) = R\left(\mathbf{A}^{\dagger}\right) = \mathbf{A}^{\dagger}(\Re^m) = \mathbf{A}^{\dagger}(R(\mathbf{A}))$. In other words, $\mathbf{A}_{/R(\mathbf{A}^T)}$ and $\mathbf{A}^{\dagger}_{/R(\mathbf{A})}$ are linear transformations such that

$$\mathbf{A}_{/R(\mathbf{A}^T)} : R\left(\mathbf{A}^T\right) \to R(\mathbf{A}) \quad \text{and} \quad \mathbf{A}^{\dagger}_{/R(\mathbf{A})} : R(\mathbf{A}) \to R\left(\mathbf{A}^T\right).$$

If $\mathcal{B} = \{\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_r\}$ and $\mathcal{B}' = \{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_r\}$ are the first r columns from $\mathbf{U} = \left(\mathbf{U}_1 \,|\, \mathbf{U}_2\right)$ and $\mathbf{V} = \left(\mathbf{V}_1 \,|\, \mathbf{V}_2\right)$ in (5.11.11), then $\mathbf{A}\mathbf{V}_1 = \mathbf{U}_1\mathbf{C}$ and $\mathbf{A}^{\dagger}\mathbf{U}_1 = \mathbf{V}_1\mathbf{C}^{-1}$ implies (recall (4.7.4)) that

$$\left[\mathbf{A}_{/R(\mathbf{A}^T)}\right]_{\mathcal{B}'\mathcal{B}} = \mathbf{C} \quad \text{and} \quad \left[\mathbf{A}^{\dagger}_{/R(\mathbf{A})}\right]_{\mathcal{B}\mathcal{B}'} = \mathbf{C}^{-1}. \tag{5.12.19}$$

If left-hand and right-hand singular vectors from the SVD (5.12.2) are used in $\mathcal{B}$ and $\mathcal{B}'$, respectively, then $\mathbf{C} = \mathbf{D} = \operatorname{diag}(\sigma_1, \ldots, \sigma_r)$. Thus (5.12.19) reveals the exact sense in which $\mathbf{A}$ and $\mathbf{A}^{\dagger}$ are "inverses." Compare these results with the analogous statements for the Drazin inverse in Example 5.10.5 on p. 399.