7 The Singular Value Decomposition

Lecture 6 discussed the least-squares approximation problem

$$\min_{\mathbf{x}\in\mathbb{R}^n} \|A\mathbf{x}-\mathbf{b}\|, \tag{1}$$

where $A \in \mathbb{R}^{m\times n}$ is a matrix with more rows than columns ($m > n$) and $\mathbf{b} \in \mathbb{R}^m$, and its solution by QR factorization of A. This lecture describes another factorization, *the singular value decomposition*, or SVD for short, which also can be used to solve least-squares problems. The SVD of a matrix is more complicated and expensive to compute than the QR factorization; however, the SVD provides more insight into the problem being solved and can be applied also in situations when QR factorization cannot, such as when the columns of A are linearly dependent. We first discuss properties and applications of the SVD. At the end of this lecture, we describe some of the computations required to determine the SVD of a matrix.

The SVD of $A \in \mathbb{R}^{m\times n}$, where we assume that $m \geq n$, is a factorization of the form

$$A = U\Sigma V^T, \tag{2}$$

where $U = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_m] \in \mathbb{R}^{m\times m}$ and $V = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n] \in \mathbb{R}^{n\times n}$ are orthogonal matrices and

$$\Sigma = \begin{bmatrix} \sigma_1 & & & \text{O} \\ & \sigma_2 & & \\ & & \ddots & \\ & & & \sigma_n \\ \text{O} & & & \end{bmatrix} \in \mathbb{R}^{m\times n}, \qquad \sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n \geq 0,$$

is a (possibly rectangular) diagonal matrix with the nonnegative diagonal entries enumerated in decreasing order. The last $m-n$ rows only contain zero entries. The σ_j are referred to as *singular values*, and the $\mathbf{u}_j$ and $\mathbf{v}_j$ as *left and right singular vectors*, respectively. The SVD also can be defined for matrices with more columns than rows; then the diagonal matrix Σ has $n-m$ trailing zero columns. This lecture is concerned with the situation when $m \geq n$.

Since the last $m-n$ rows of Σ only contain zeros, the decomposition (2) also can be written in the form

$$A = \sum_{j=1}^{n} \sigma_j \mathbf{u}_j \mathbf{v}_j^T. \tag{3}$$

This representation only uses the first n columns of U.

[0] Version December 4, 2013

7.1 The range and null space of a matrix via its SVD

Some singular values of A may vanish. Assume that A has ℓ positive singular values, i.e.,

$$\sigma_1 \geq \ldots \geq \sigma_\ell > \sigma_{\ell+1} = \ldots = \sigma_n = 0. \tag{4}$$

The range of the matrix A easily can be determined from its singular value decomposition. We have

$$\text{range}(A) = \{A\mathbf{x} : \mathbf{x} \in \mathbb{R}^n\} = \{U\Sigma V^T\mathbf{x} : \mathbf{x} \in \mathbb{R}^n\} = \{U\Sigma\mathbf{y} : \mathbf{y} \in \mathbb{R}^n\}, \tag{5}$$

where

$$\mathbf{y} = [y_1, y_2, \ldots, y_n]^T = V^T\mathbf{x}.$$

Let $c_j = \sigma_j y_j$ for $j = 1, 2, \ldots, \ell$. Then the right-hand side of (5) can be written as

$$\text{range}(A) = \{U\Sigma\mathbf{y} : \mathbf{y} \in \mathbb{R}^n\} = \left\{\sum_{j=1}^{\ell} \mathbf{u}_j c_j : c_j \in \mathbb{R}\right\} = \text{span}\{\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_\ell\}. \tag{6}$$

Hence, range(A) is spanned by the columns of U that are associated with positive singular values in the representation (3). These columns form an orthonormal basis for the range.

Recall from Lecture 1 that the dimension of range(A) is referred to as the rank of A. It is denoted by rank(A). It follows from (6) that rank(A) $= \ell$, i.e., the rank is the number of positive singular values.

Assume that the columns of the matrix A are linearly dependent. Then A has a nontrivial null space null(A); see Lecture 1 for the definition of null(A). The null space can be expressed in terms of the last columns of the matrix V. We have

$$\text{null}(A) = \{\mathbf{x} \in \mathbb{R}^n : A\mathbf{x} = \mathbf{0}\} = \{\mathbf{x} \in \mathbb{R}^n : U\Sigma V^T\mathbf{x} = \mathbf{0}\} = \{\mathbf{x} \in \mathbb{R}^n : \Sigma V^T\mathbf{x} = \mathbf{0}\},$$

where the last expression is obtained by multiplying the equation $U\Sigma V^T\mathbf{x} = \mathbf{0}$ by U^T from the left.

Any vector $\mathbf{x} \in \mathbb{R}^n$ can be written as a linear combination of the columns $\mathbf{v}_j$ of the matrix V, i.e., $\mathbf{x} = \sum_{j=1}^n \mathbf{v}_j c_j$ for certain coefficients c_j. Substituting this expression into

$$\Sigma V^T\mathbf{x} = \mathbf{0}$$

and using the orthogonality of the columns $\mathbf{v}_j$ gives the equation

$$[\sigma_1 c_1, \sigma_2 c_2, \ldots, \sigma_\ell c_\ell, \sigma_{\ell+1} c_{\ell+1}, \ldots, \sigma_n c_n]^T = \mathbf{0}.$$

Since $\sigma_j > 0$ for $1 \leq j \leq \ell$, it follows that $c_j = 0$ for these index values. Moreover, $\sigma_j = 0$ for $\ell < j \leq n$ implies that the coefficients c_j for these j-values are arbitrary. We conclude that

$$\text{null}(A) = \left\{\mathbf{x} = \sum_{j=\ell+1}^{n} \mathbf{v}_j c_j, c_j \in \mathbb{R}\right\} = \text{span}\{\mathbf{v}_{\ell+1}, \mathbf{v}_{\ell+2}, \ldots, \mathbf{v}_n\}. \tag{7}$$

Thus, null(A) is spanned by the columns of V associated with vanishing singular values. If all singular values are positive, then null(A) = $\{\mathbf{0}\}$.

The singular value decomposition of A^T is given by

$$A^T = (U\Sigma V^T)^T = V\Sigma^T U^T$$

and analogously to equations (6) and (7), we obtain

$$\begin{aligned}\text{range}(A^T) &= \{V\Sigma^T U^T\mathbf{x} : \mathbf{x} \in \mathbb{R}^m\} = \{V\Sigma^T\mathbf{y} : \mathbf{y} \in \mathbb{R}^m\} \\ &= \left\{\sum_{j=1}^{\ell} \mathbf{v}_j c_j : c_j \in \mathbb{R}\right\} = \text{span}\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_\ell\}\end{aligned} \tag{8}$$

and

$$\begin{aligned}\text{null}(A^T) &= \{\mathbf{x} \in \mathbb{R}^m : A^T\mathbf{x} = \mathbf{0}\} = \{\mathbf{x} \in \mathbb{R}^m : \Sigma^T U^T\mathbf{x} = \mathbf{0}\} \\ &= \left\{\sum_{j=\ell+1}^{m} \mathbf{u}_j c_j, c_j \in \mathbb{R}\right\} = \text{span}\{\mathbf{u}_{\ell+1}, \mathbf{u}_{\ell+2}, \ldots, \mathbf{u}_m\}.\end{aligned} \tag{9}$$

Comparing equations (6) and (9) shows that the sets range(A) and null(A^T) are orthogonal, i.e., any vector in range(A) is orthogonal to any vector in null(A^T). Moreover, the union of these spaces is spanned by all the columns of U, i.e., the union is $\mathbb{R}^m$. This result is sometimes written as

$$\text{range}(A) \oplus \text{null}(A^T) = \mathbb{R}^m. \tag{10}$$

Similarly, a comparison of (7) and (8) shows that the sets null(A) and range(A^T) are orthogonal, and that the union of these sets is $\mathbb{R}^n$. This property is sometimes expressed as

$$\text{null}(A) \oplus \text{range}(A^T) = \mathbb{R}^n. \tag{11}$$

The properties (10) and (11) are shown in different, more complicated, ways in standard Linear Algebra courses. Our reason for discussing these results here is to illustrate that they follow quite easily from the SVD of A.

Exercise 7.1

What is the singular value decomposition of the matrix

$$A = \begin{bmatrix} 2 & 0 \\ 0 & 4 \end{bmatrix}?$$

Determine the solution with a paper and pencil! □

Exercise 7.2

What is the singular value decomposition of the matrix

$$A = \begin{bmatrix} 2 & 0 \\ 0 & -4 \end{bmatrix}?$$

Determine the solution with a paper and pencil! □

Exercise 7.3

Compute the SVD of the matrix

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 3 & 4 \\ 4 & 5 & 6 \\ 1 & 1 & 1 \end{bmatrix}.$$

What are the range and null space of A? You may use the MATLAB/Octave commad `svd` to determine the SVD. □

7.2 The SVD applied to matrix norm computations

Recall from Lecture 1 the definition of the matrix norm induced by the Euclidean vector norm,

$$\|A\| = \max_{\|\mathbf{x}\|=1} \|A\mathbf{x}\|. \tag{12}$$

Substituting the SVD of A into the right-hand side of (12) yields

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

Using that the Euclidean vector norm is invariant under multiplication by an orthogonal matrix allows us to discard the matrix U above. Moreover, letting $\mathbf{y} = V^T\mathbf{x}$ and observing that $\|\mathbf{y}\| = \|\mathbf{x}\|$ (since V^T is orthogonal), it follows that

$$\|A\| = \max_{\|\mathbf{x}\|=1} \|\Sigma V^T\mathbf{x}\| = \max_{\|\mathbf{y}\|=1} \|\Sigma\mathbf{y}\| = \sigma_1.$$

The last equality is obtained by explicitly writing up the norm of the vector $\Sigma\mathbf{y}$ and determining when the norm is maximal. We have shown that

$$\|A\| = \sigma_1. \tag{13}$$

In words, *the norm of a matrix is the largest singular value of the matrix.*

Let the matrix $A \in \mathbb{R}^{n\times n}$ be invertible. We will use the above result to determine the norm of A^{-1}. Consider the SVD (2) of A. Since A is square, so are the matrices U, Σ, and V. Moreover, since A has is invertable, its smallest singular value, σ_n, is positive. This follows for instance from the observation that if $\sigma_n = 0$, then the right singular vector $\mathbf{v}_n$ is in null(A). However, we know that null$(A) = \{\mathbf{0}\}$. Thus, the matrix Σ is invertible and we obtain

$$A^{-1} = (U\Sigma V^T)^{-1} = V\Sigma^{-1}U^T. \tag{14}$$

The right-hand side is the SVD of A^{-1} up to a reordering of the columns of U and V; see Exercise 7.4. Reordering does not affect the size of the the singular values, which are the diagonal entries of Σ^{-1}. Thus, A^{-1} has the singular values

$$\sigma_n^{-1} \geq \sigma_{n-1}^{-1} \geq \ldots \geq \sigma_1^{-1}.$$

The largest singular value of A^{-1} is σ_n^{-1}. Therefore,

$$\|A^{-1}\| = \sigma_n^{-1}. \tag{15}$$

In words, *the norm of the inverse of a matrix is the reciprocal of the smallest singular value of the matrix.*

Exercise 7.4

Express the SVD of A^{-1} in terms of the matrices U, Σ, and V in the SVD of A; cf. (14). Note that the columns of U and V have to be reordered suitably. □

7.3 Approximations of A determined by its SVD

Our starting point is the representation (3) of A. Assume that the singular values satisfy (4). Then each term $\sigma_j \mathbf{u}_j \mathbf{v}_j^T$ with $1 \leq j \leq \ell$ in (3) is a rank-one matrix. Introduce, for $1 \leq k \leq \ell$, the matrices

$$A_k = \sum_{j=1}^{k} \sigma_j \mathbf{u}_j \mathbf{v}_j^T. \tag{16}$$

Thus, rank$(A_k) = k$. We also will require the diagonal matrices Σ_k obtained by setting the singular values $\sigma_{k+1}, \sigma_{k+2}, \ldots, \sigma_\ell$ to zero in the matrix Σ.

Let $1 \leq k \leq \ell$. The difference $A - A_k$ is small if the singular values $\sigma_{k+1}, \sigma_{k+2}, \ldots, \sigma_\ell$ are small. We have

$$A - A_k = \sum_{j=k+1}^{\ell} \sigma_j \mathbf{u}_j \mathbf{v}_j^T = U\Sigma V^T - U\Sigma_k V^T$$

and, therefore,

$$\|A - A_k\| = \|U\Sigma V^T - U\Sigma_k V^T\| = \|\Sigma - \Sigma_k\| = \sigma_{k+1}. \tag{17}$$

In the special case, when $k = n$, we define $\sigma_{n+1} = 0$.

The equality (17) shows that if k is chosen so that σ_{k+1} is small, then A_k is an accurate approximation of A. In some applications with large matrices A, we may choose k to be much smaller than ℓ and still obtain an acceptable approximation of A. Instead of storing the matrix A, it therefore suffices to store the first k left and right singular vectors and singular values in the sum (16). This illustrates how the SVD can be applied to *data compression*.

We have the following remarkable result:

$$\|A - A_k\| = \min_{\substack{B \in \mathbb{R}^{m \times n} \\ \mathrm{rank}(B) \leq k}} \|A - B\| = \sigma_{k+1}.$$

Thus, the matrix A_k is the best possible rank-k approximation of A. The proof is not very difficult, but outside the scope of this course.

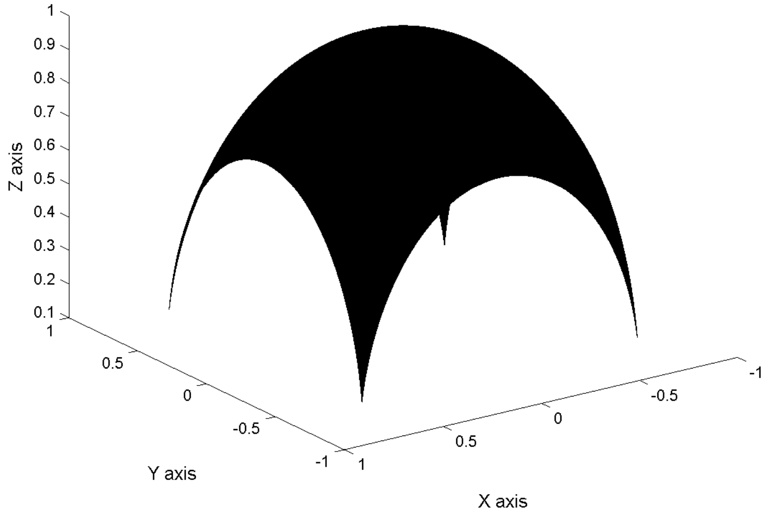

Figure 1: The surface represented by the function of Example 7.1.

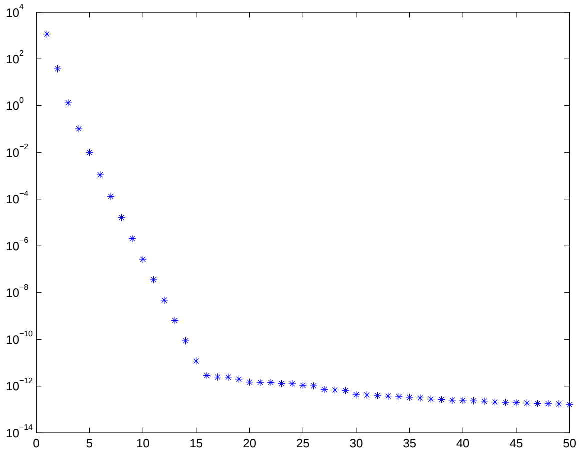

Figure 2: The 50 largest singular values of the 1401 × 1401 matrix A of Example 7.1.

Example 7.1

Let the matrix $A = [a_{ij}] \in \mathbb{R}^{1401\times 1401}$ represent a discretization of the surface

$$z = \sqrt{1 - x^2 - y^2}, \quad -0.7 \le x, y \le 0.7.$$

Specifically, let

$$a_{ij} = \sqrt{1 - x_i^2 - y_j^2}, \quad x_i = y_i = -0.7 + \Delta(i-1), \quad \Delta = 0.001, \quad 1 \le i, j \le 1401.$$

Figure 1 shows a plot of the surface. The first 50 singular values of the matrix A are displayed by Figure 2. These singular values are seen to decay to zero rapidly, which suggests that an accurate approximation of the matrix can be determined by a sum of the form (16) with few terms. For instance $\sigma_1 = 1.1 \cdot 10^3$, $\sigma_5 = 1.0 \cdot 10^{-2}$, and $\sigma_6 = 1.1 \cdot 10^{-3}$. It follows from (17) that $\|A - A_5\| = 1.1 \cdot 10^{-3}$. Therefore the surface represented by A_5 is indistinguishable from the surface represented by A with the resolution of Figure 1. □

Exercise 7.5

Each term $\sigma_j \mathbf{u}_j \mathbf{v}_j^T$ in the right-hand side of (16) is a matrix of rank one. Determine its norm. □

Exercise 7.6

Show equation (17). □

Exercise 7.7

What is the best rank-2 approximation A_2 of the matrix in Exercise 7.3? What is $\|A - A_2\|$? □

Exercise 7.8

Reproduce the graphs of Figure 1 and 2. Use the MATLAB commands meshgrid and surface for the former and semilogy for the latter. The MATLAB command hold also may be useful. □

Exercise 7.9

Investigate whether a 501×501 matrix that represents the surface $\exp(-x^2 - 2y^2)$, $-1 \leq x, y \leq 1$, can be represented by a matrix of the form (16) of low rank k. Use the MATLAB commands `meshgrid` and `surface`. Verify experimentally that (17) holds. □

7.4 Computation with the SVD

We turn to the application of the SVD to the solution of the least-squares problem (1). Substitute the SVD (2) into (1), and introduce the vectors

$$\mathbf{y} = [y_1, y_2, \ldots, y_n]^T = V^T\mathbf{x}, \qquad \hat{\mathbf{b}} = [\hat{b}_1, \hat{b}_2, \ldots, \hat{b}_m]^T = U^T\mathbf{b}.$$

We obtain, by using the orthogonality of U, that

$$\min_{\mathbf{x}\in\mathbb{R}^n} \|A\mathbf{x} - \mathbf{b}\| = \min_{\mathbf{x}\in\mathbb{R}^n} \|U\Sigma V^T\mathbf{x} - \mathbf{b}\| = \min_{\mathbf{x}\in\mathbb{R}^n} \|\Sigma V^T\mathbf{x} - U^T\mathbf{b}\| = \min_{\mathbf{y}\in\mathbb{R}^n} \|\Sigma\mathbf{y} - \hat{\mathbf{b}}\|. \tag{18}$$

The least-squares problem on the right-hand side of (18) involves a diagonal matrix and can be solved explicitly. Assume that the singular values satisfy (4). Then we have

$$\min_{\mathbf{y}\in\mathbb{R}^n} \|\Sigma\mathbf{y} - \hat{\mathbf{b}}\|^2 = \min_{\mathbf{y}\in\mathbb{R}^n} \sum_{j=1}^{n} (\sigma_j y_j - \hat{b}_j)^2 = \min_{\mathbf{y}\in\mathbb{R}^n} \left\{ \sum_{j=1}^{\ell} (\sigma_j y_j - \hat{b}_j)^2 + \sum_{j=\ell+1}^{n} \hat{b}_j^2 \right\}, \tag{19}$$

where the last equality follows from the fact that $\sigma_{\ell+1} = \sigma_{\ell+2} = \ldots = \sigma_n = 0$. If $\text{null}(A) = \{\mathbf{0}\}$, then $\sigma_n > 0$ and $\ell = n$. The last sum vanishes in this situation.

The least-squares problem (19) has the solution

$$y_j = \frac{\hat{b}_j}{\sigma_j}, \qquad 1 \leq j \leq \ell, \tag{20}$$

$$y_j \quad \text{arbitrary}, \qquad \ell < j \leq n. \tag{21}$$

In many applications one sets the arbitrary components of the vector $\mathbf{y} = [y_1, y_2, \ldots, y_n]^T$ to zero. This gives the least-squares solution $\mathbf{y}$ of minimal norm. The associated solution $\mathbf{x} = V\mathbf{y}$ of the least-squares problem also is of minimal norm, since the norm is invariant under multiplication by the orthogonal matrix V. This solution of the least-squares problem (1) is referred to as the *minimal norm least-squares solution.* It is unique.

We remark that when A is of rank less than n, the solution method of Lecture 6 typically will not give the minimal norm least-squares solution. The upper triangular matrix in the QR factorization of A will be singular; it will have at least one zero entry on the diagonal. This shows that the least-squares problem does not have a unique solution; however, it is difficult to determine the least-squares solution of minimal norm using the QR factorization of A.

When some positive singular values are "tiny," we see from (20) that the computed solution may have large components. Assume that the smallest positive singular value is much smaller than the largest singular value. Then it may be meaningful to set the former to zero. The representation (3) of A shows that setting a tiny singular value to zero induces a tiny change in A; a term, say, $\sigma_n \mathbf{u}_n \mathbf{v}_n^T$ of tiny norm σ_n is replaced by the zero matrix. This small modification of A will not affect the minimum value (19) significantly. However, the modified problem may have a solution of much smaller norm than the original problem and therefore be more meaningful in the context of an application. The following section illustrates this for polynomial approximation.

Exercise 7.10

Let A be the matrix of Exercise 7.3 and let $\mathbf{b} = [1, 1, 1, 1]^T$. Determine the least-squares solution of

$$\min_{\mathbf{x}\in\mathbb{R}^3} \|A\mathbf{x} - \mathbf{b}\|$$

by using the MATLAB command $\mathbf{x} = A\backslash\mathbf{b}$. Is this the minimal norm least-squares solution? □

7.5 An application to polynomial approximation

We consider the problem of approximating the function $f(t) = \exp(t)$ on the interval $[0, 1]$ by a polynomial of degree at most 10 by using 11 function values at equidistant points. The available function values are contaminated by error, which, for instance, may stem from poor software for evaluating the exponential function.

Denote the exact function values, which are assumed not to be available, by $f_j = \exp(t_j)$, $1 \leq j \leq 11$, where the nodes are defined by $t_j = (j-1)/10$, $1 \leq j \leq 11$. Thus, the nodes are equidistant on the interval $[0, 1]$. Let

$$\tilde{f}_j = f_j + \eta_j, \qquad 1 \leq j \leq 11,$$

denote the available approximations of f_j. Here η_j is the error in $\tilde{f}_j$; the η_j are normally distributed random numbers with zero mean and their variance (scaling) corresponds to a relative error of 1% in the following sense: Introduce the vectors

$$\mathbf{f} = [f_1, f_2, \ldots, f_{11}]^T, \qquad \mathbf{n} = [\eta_1, \eta_2, \ldots, \eta_{11}]^T.$$

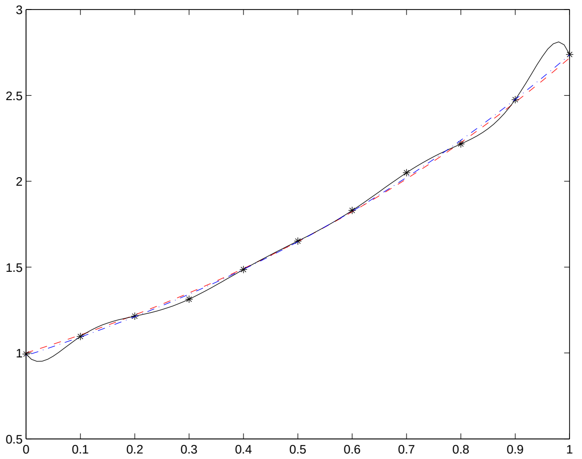

Figure 3: The function f and the polynomial that interpolates f at 11 equidistant nodes (red dashed graph), the polynomial that interpolates the error-contaminated approximations f (black solid graph), the interpolation points (marked by $*$), and the polynomial obtained by computing the SVD of the Vandermonde matrix and setting the singular values σ_j to zero for $j \geq 5$ (blue dash-dotted graph).

The variance (scaling) of $\mathbf{n}$ is such that $\|\mathbf{n}\|/\|\mathbf{f}\| = 0.01$. We refer to the vector $\mathbf{n}$ as "noise." It easily can be generated in MATLAB in the following way:

$$\mathbf{n} = \mathsf{randn}(11,1); \qquad \mathbf{n} = \mathbf{n}/\mathsf{norm}(\mathbf{n}); \qquad \mathbf{n} = \mathbf{n} * \mathsf{norm}(\mathbf{f}) * 0.01;$$

We first compute a polynomial

$$p(t) = c_1 + c_2 t + c_3 t^2 + \ldots + c_{11} t^{10}, \tag{22}$$

which attains the exact function values f_j at the nodes t_j. This polynomial will be used for comparison. Thus, p satisfies

$$p(t_j) = f(t_j), \qquad 1 \leq j \leq 11. \tag{23}$$

The polynomial p is said to *interpolate* f at the nodes t_j. The latter are also referred to as *interpolation points*.

Substituting (22) into (23) yields a linear system of equations

$$A\mathbf{c} = \mathbf{f}, \tag{24}$$

where

$$A = \begin{bmatrix} 1 & t_1 & t_1^2 & \dots & t_1^{10} \\ 1 & t_2 & t_2^2 & \dots & t_2^{10} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & t_{11} & t_{11}^2 & \dots & t_{11}^{10} \end{bmatrix}, \qquad \mathbf{c} = [c_1, c_2, \dots, c_{11}]^T. \tag{25}$$

Matrices of the above form are known as *Vandermonde matrices.* They can be shown to be nonsingular when all nodes t_j are distinct. For instance, 2×2 Vandermonde matrices satisfy

$$\det \begin{bmatrix} 1 & t_1 \\ 1 & t_2 \end{bmatrix} = t_2 - t_1 \neq 0.$$

The nonsingularity of $n \times n$ Vandermonde matrices with distinct nodes can be shown by induction and guarantees the existence of a unique interpolation polynomial. In particular, the linear system of equations (24) has a unique solution.

We measure the approximation error using a discrete uniform norm that evaluates $f - p$ at 101 equidistant nodes in $[0, 1]$:

$$\|f - p\|_\infty = \max_{1 \leq j \leq 101} \left| f\left(\frac{j-1}{100}\right) - p\left(\frac{j-1}{100}\right) \right|.$$

This yields $\|f - p\|_\infty = 1.8 \cdot 10^{-13}$. Thus, when using exact function values, the polynomial determined by solving the Vandermonde system (24) approximates f on the whole interval $[0, 1]$ very accurately. The dashed red graph of Figure 3 depicts this polynomial as well as the function f; the graphs are too close to distinguish.

We now determine the polynomial

$$\tilde{p}(t) = \tilde{c}_1 + \tilde{c}_2 t + \tilde{c}_3 t^2 + \ldots + \tilde{c}_{11} t^{10} \tag{26}$$

that interpolates the contaminated function values $\tilde{f}_j$, i.e.,

$$\tilde{p}(t_j) = \tilde{f}_j, \qquad 1 \leq j \leq 11. \tag{27}$$

Substituting (26) into (27) gives the linear system of equations

$$A\tilde{\mathbf{c}} = \tilde{\mathbf{f}}, \tag{28}$$

where $\tilde{\mathbf{f}} = [\tilde{f}_1, \tilde{f}_2, \dots, \tilde{f}_{11}]^T$ and $\tilde{\mathbf{c}} = [\tilde{c}_1, \tilde{c}_2, \dots, \tilde{c}_{11}]^T$, for the coefficients of $\tilde{c}_j$ of $\tilde{p}$. The black continuous graph of Figure 3 displays $\tilde{p}$. The points $\{(t_j, \tilde{f}_j)\}_{j=1}^{11}$ are marked by $*$.

The graph shows the polynomial $\tilde{p}$ to be a poor approximation of f. Moreover, the error in $\tilde{p}$ is seen to be larger between the nodes t_j than at the nodes. We have

$$\|f - \tilde{p}\|_\infty = 1.6 \cdot 10^{-1}$$

and

$$\max_{1\leq j\leq 11} |f_j - \tilde{f}_j| = 3.7 \cdot 10^{-2}.$$

There are two fairly simple approaches to determining a better polynomial approximant from the available data: i) fit a polynomial of lower degree using the least-squares method, or ii) compute the SVD of the Vandermonde matrix and set a few of the smallest singular values to zero in order to avoid division of by tiny numbers, cf. (20), and the possible resulting error propagation. We will use the latter approach.

Consider the SVD of the Vandermonde matrix (25),

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

with the orthogonal matrices

$$U = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_{11}] \in \mathbb{R}^{11\times 11}, \qquad V = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_{11}] \in \mathbb{R}^{11\times 11}$$

and the diagonal matrix

$$\Sigma = \text{diag}[\sigma_1, \sigma_2, \ldots, \sigma_{11}], \qquad \sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_{11} > 0.$$

Using this decomposition, the solution of (28) can be written as

$$\tilde{\mathbf{c}} = A^{-1}\tilde{\mathbf{f}} = V\Sigma^{-1}U^T\tilde{\mathbf{f}} = \sum_{j=1}^{11} \frac{\hat{\tilde{f}}_j}{\sigma_j}\mathbf{v}_j, \tag{29}$$

with

$$\hat{\tilde{\mathbf{f}}} = [\hat{\tilde{f}}_1, \hat{\tilde{f}}_2, \ldots, \hat{\tilde{f}}_{11}]^T := U^T\tilde{\mathbf{f}}.$$

Figure 4 displays in logarithmic scale the singular values of A. The singular values can be seen to decrease faster than exponentially with increasing index j (exponential decrease would result in a straight line with negative slope).

The presence of tiny singular values implies that we divide by tiny numbers in the sum in (29). In order to gain further understanding of whether this causes error propagation, we also plot the the numerators $\hat{\tilde{f}}_j$ of this sum.

Figure 5 shows in logarithmic scale the magnitude of the components of the vectors $\hat{\mathbf{f}} = U^T\mathbf{f}$ and $\hat{\tilde{\mathbf{f}}} = U^T\tilde{\mathbf{f}}$. The magnitude of the components of the former vector is seen decrease to zero much more rapidly with increasing index j. The difference in magnitude of the components of $\hat{\mathbf{f}}$ and $\hat{\tilde{\mathbf{f}}}$ depends on the error in the latter vector. The error in the components of $\hat{\tilde{\mathbf{f}}}$ with large index is divided by small singular values. This amplifies the influence of the errors in the function values $\tilde{f}_j$ on the computed polynomial coefficients.

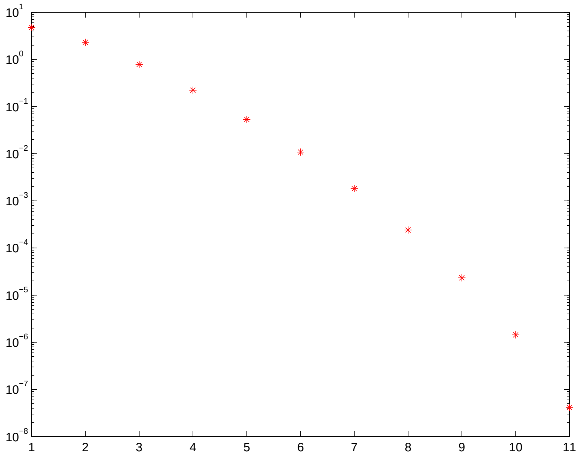

Figure 4: Singular values σ_j of the Vandermonde matrix A versus j.

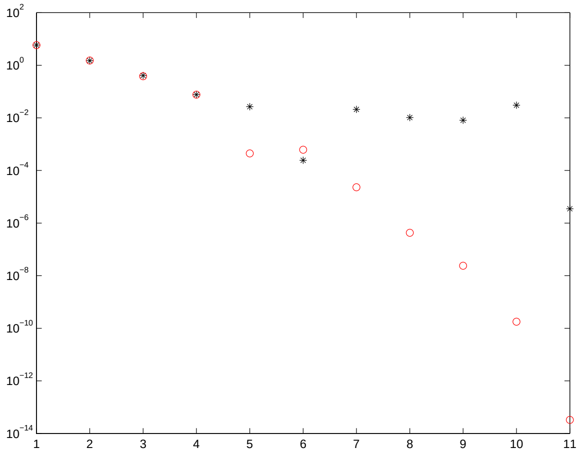

Figure 5: Magnitude of the components $\hat{\tilde{f}}_j$ of the vector $\hat{\tilde{\mathbf{f}}}$ in the numerator in the sum in (29) versus j (black $*$) and magnitude of the components of the error-free vector $\hat{f}_j$ versus j (red circles). The magnitude of the latter components is seen to decrease to zero faster.

A remedy for this problem is to ignore the last terms with tiny singular values in the sum (29).

ℓ	$\|f-\tilde{p}^{(\ell)}\|_\infty$
11	0.162
10	0.136
9	0.035
8	0.026
7	0.025
6	0.022
5	0.020
4	0.019
3	0.020
2	0.134
1	0.985

Table 1: Approximation error for polynomials $\tilde{p}^{(\ell)}$ determined by the entries of the vector $\tilde{\mathbf{c}}^{(\ell)}$ given by (30).

We therefore determine the approximate solution

$$\tilde{\mathbf{c}}^{(\ell)} = \sum_{j=1}^{\ell} \frac{\hat{\tilde{f}}_j}{\sigma_j} \mathbf{v}_j, \tag{30}$$

to the linear system of equations (28) for some $\ell < n$. Choosing the number of terms ℓ is equivalent to replacing the Vandermonde matrix A by a nearby matrix of rank ℓ.

Let $\tilde{p}^{(\ell)}$ denote the polynomial determined by the coefficients in the solution vector $\mathbf{c}^{(\ell)}$. Table 1 shows how well the $\tilde{p}^{(\ell)}$ approximate f for different values of ℓ. The table shows the error to be the smallest for $\ell = 4$, i.e., when the Vandermonde matrix A is approximated by a matrix of rank 4. The blue dash-dotted graph of Figure 3 displays $\tilde{p}^{(4)}$. This graph is much closer to the graph of f than the graph for the interpolating polynomial $\tilde{p}$.

These computations illustrates that the SVD may be used to determine accurate polynomial approximants when interpolation does not. The determination of the value of ℓ in (30) that yields the polynomial that best approximates f is often is not easy when f is not explicitly known. A graph for the components of the vector $\hat{\tilde{\mathbf{f}}}$ often is helpful for choosing an appropriate ℓ. We may, for instance, choose ℓ small enough to exclude terms associated with coefficients $\hat{\tilde{f}}_j$ that do not decrease in magnitude. This approach applied to the present problem gives the value $\ell = 6$; see Figure 5. This value is not optimal; however, Table 1 shows the error for the polynomial $\tilde{p}^{(6)}$ to be almost as small as for the best polynomial $\tilde{p}^{(4)}$.

Exercise 7.11

Carry out the computations of Subsection 7.5 with relative noise $\|\mathbf{n}\|/\|\mathbf{f}\| = 0.1$ and $\|\mathbf{n}\|/\|\mathbf{f}\| = 0.001$. Determine analogs of Table 1. What are the optimal values of ℓ for these noise levels? □

Exercise 7.12

Discuss how one might be able to determine a suitable value of ℓ in Exercise 7.11 when the errors $\tilde{f}_j - f_j$ are not known? □

Exercise 7.13

Solve the polynomial approximation problem of Section 7.5 by using QR factorization of the matrix A. Polynomials of lower degree can be fitted by removing the last column(s) of A. How does the QR factorization change when the last column of A is removed? Compare the quality of computed polynomial of different degrees. See Example 6.5 of Lecture 6 for more details on polynomial least-squares approximation. □

7.6 Computation of the SVD

The computation of the SVD (2) of an $m \times n$ matrix A with $m \geq n$ requires about $10mn^2$ arithmetic floating point operations and therefore is quite expensive for large matrices. The first phase of the computation is to reduce A to a bidiagonal matrix using Householder transformations. The first Householder transformation, H_1, is applied from the left-hand side to create zeros below the diagonal in the first column, just like the first step of QR factorization. Let for notational simplicity A be a 5×4 matrix and denote entries that may be nonzero by $*$. Then

$$H_1 A = \begin{bmatrix} * & * & * & * \\ & * & * & * \\ & * & * & * \\ & * & * & * \\ & * & * & * \end{bmatrix}.$$

We next apply the Householder matrix H_2 from the right-hand side to create zeros in all but the first two entries of row one. This yields

$$H_1 A H_2 = \begin{bmatrix} * & * & & \\ & * & * & * \\ & * & * & * \\ & * & * & * \\ & * & * & * \end{bmatrix}.$$

We refer to the matrix H_2 as a Householder matrix, but strictly speaking it is of the form $\hat{H}^{(2)}$ in formula (11) of Lecture 6. Note that one cannot also zero out the $(1,2)$-entry of H_1A without risking fill-in of nonzero entries in the first column. Therefore, we determine a bidiagonal matrix instead of a diagonal one.

Another Householder matrix, H_3, is applied from the left-hand side to set all elements below the diagonal in the second column to zero. This matrix is of the form $\hat{H}^{(3)}$ in formula (13) of Lecture 6. The matrix H_3 does not affect the first row and first column of H_1AH_2. Thus,

$$H_3H_1AH_2 = \begin{bmatrix} * & * & & \\ & * & * & * \\ & & * & * \\ & & * & * \\ & & * & * \end{bmatrix}.$$

We now apply the Householder matrix H_4 from the right-hand side to set the $(2,3)$-entry to zero and the Householder matrix H_5 from the left-hand side to zero out all entries below the diagonal in the third column of the resulting matrix. This gives the matrix

$$H_5H_3H_1AH_2H_4 = \begin{bmatrix} * & * & & \\ & * & * & \\ & & * & * \\ & & & * \\ & & & * \end{bmatrix},$$

which is almost bidiagonal. It remains to apply a Householder matrix, H_6, from the left-hand side. The matrix H_6 acts on the last two rows of $H_5H_3H_1AH_2H_4$ and is designed to set the $(5,4)$-entry to zero. This gives the desired bidiagonal matrix

$$H_6H_5H_3H_1AH_2H_4 = \begin{bmatrix} * & * & & \\ & * & * & \\ & & * & * \\ & & & * \\ & & & \end{bmatrix}. \tag{31}$$

The next phase of the computation of the SVD of A is to transform the bidiagonal matrix (31) to diagonal form. This will give Σ. The transformation of the matrix (31) to diagonal form is carried out by application of orthogonal matrices from the left-hand side and from the right-hand side. This is an iterative process, which requires the application of quite a large number of orthogonal matrices. The product of all orthogonal matrices applied from the left-hand side, including $H_6H_5H_3H_1$, makes up the matrix U^T in (2). Similarly, the product of all orthogonal matrices applied from the right-hand side, including H_2H_4, makes up the matrix V.

This method for computing the SVD of a matrix is closely related to methods for determining eigenvalues and eigenvectors of a symmetric matrix. The latter methods will be discussed in a later lecture, and we will return to the computation of the SVD at that time.

Exercise 7.14

Reduce the matrix

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 3 & 4 \\ 4 & 5 & 6 \\ 1 & 7 & 8 \end{bmatrix}$$

to bidiagonal form by application of Householder matrices. □

Exercise 7.15

The reduction to bidiagonal form can be sped up when the matrix A has many more rows than columns by first computing its QR factorization and then reducing the triangular matrix to bidiagonal form as descibed above.

Assume that the QR factorization of the matrix $A \in \mathbb{R}^{10\times 3}$ is available. Describe how a bidiagonal matrix can be determined from the upper triangular matrix in the QR factorization by application of only two Householder-type matrices. What is the analog of (31)? □

Exercise 7.16

What is the computational effort required to compute the bidiagonal form (31)? You may assume that $m = n$. □

Exercise 7.17

What is the computational effort required to compute the bidiagonal form (31) when $m \gg n$? The computations can be arranged as in Exercise 7.15. □