Chapter 7

Reduced-Rank Least Squares Models

Consider a linear model

$$b = Ax + \eta, \qquad A \in \mathbb{R}^{m \times n},$$

where η is random noise and A and b are given. If one chooses to determine x by minimizing the Euclidean norm of the residual $b - Ax$, then one has a linear least squares problem

$$\min_x \|Ax - b\|_2. \tag{7.1}$$

In some situations the actual solution x itself is not the primary object of interest, but rather it is an auxiliary, intermediate variable. This is the case, e.g., in certain classification methods, when the norm of the residual is the interesting quantity; see Chapter 10.

Least squares prediction is another area in which the solution x is an intermediate quantity and is not interesting in itself (except that it should be robust and reliable in a numerical sense). The columns of A consist of observations of *explanatory variables*, which are used to explain the variation of a *dependent variable* b. In this context it is essential that the variation of b is well explained by an approximate solution $\hat{x}$, in the sense that the *relative residual* $\|A\hat{x} - b\|_2/\|b\|_2$ should be rather small. Given $\hat{x}$ and a new row vector a_{new}^T of observations of the explanatory variables, one can predict the corresponding value of the dependent variable:

$$b_{\text{predicted}} = a_{\text{new}}^T \hat{x}. \tag{7.2}$$

Often in this context it is not necessary, or even desirable, to find the solution that actually minimizes the residual in (7.1). For instance, in prediction it is common that several of the explanatory variables are (almost) linearly dependent. Therefore the matrix A is often very ill-conditioned, and the least squares solution is highly influenced by measurement errors and floating point roundoff errors.

Example 7.1. The MATLAB script

```
A=[1 0
   1 1
   1 1];
B=[A A*[1;0.5]+1e-7*randn(3,1)];
b=B*[1;1;1]+1e-4*randn(3,1);
x=B\b
```

creates a matrix B, whose third column is almost a linear combination of the other two. The matrix is quite ill-conditioned: the condition number is $\kappa_2(B) \approx 5.969{\cdot}10^7$. The script gives the least squares solution

```
x = -805.95
    -402.47
     807.95
```

This approximate solution explains the variation of the dependent variable very well, as the residual is small:

```
resn = norm(B*x-b)/norm(b) = 1.9725e-14
```

However, because the components of the solution are large, there will be considerable cancellation (see Section 1.5) in the evaluation of (7.2), which leads to numerical errors. Furthermore, in an application it may be very difficult to interpret such a solution.

The large deviation of the least squares solution from the vector that was used to construct the right-hand side is due to the fact that the *numerical rank* of the matrix is two. (The singular values are 3.1705, 0.6691, and $8.4425 \cdot 10^{-8}$.) In view of this, it is reasonable to accept the approximate solution

```
xt = 0.7776
     0.8891
     1.2222
```

obtained using a truncated SVD

$$x_{\rm tsvd} = \sum_{i=1}^{2} \frac{u_i^T b}{\sigma_i} v_i$$

(cf. Section 7.1). This solution candidate has residual norm $1.2 \cdot 10^{-5}$, which is of the same order of magnitude as the perturbation of the right-hand side. Such a solution vector may be much better for prediction, as the cancellation in the evaluation of (7.2) is much smaller or is eliminated completely (depending on the vector $a_{\rm new}$). ∎

To reduce the ill-conditioning of the problem, and thus make the solution less sensitive to perturbations of the data, one sometimes introduces an approximate orthogonal basis of low dimension in $\mathbb{R}^n$, where the solution x lives. Let the basis vectors be $\begin{pmatrix} z_1 & z_2 & \dots & z_k \end{pmatrix} =: Z_k$, for some (small) value of k. Then to determine the coordinates of the solution in terms of the approximate basis, we make the *ansatz* $x = Z_k y$ in the least squares problem and solve

$$\min_y \|AZ_k y - b\|_2. \tag{7.3}$$

This is a least squares problem corresponding to a *reduced-rank model.* In the following two sections we describe two methods for determining such a matrix of basis vectors. The first is based on the SVD of the data matrix A. The second method is a *Krylov subspace method*, in which the right-hand side influences the choice of basis.

7.1 Truncated SVD: Principal Component Regression

Assume that the data matrix has the SVD

$$A = U\Sigma V^T = \sum_{i=1}^{r} \sigma_i u_i v_i^T,$$

where r is the rank of A (note that we allow $m \geq n$ or $m \leq n$). The minimum norm solution (see Section 6.7) of the least squares problem (7.1) is

$$x = \sum_{i=1}^{r} \frac{u_i^T b}{\sigma_i} v_i. \tag{7.4}$$

As the SVD "orders the variation" of the data matrix A starting with the dominating direction, we see that the terms in the sum (7.4) are also organized in this way: the first term is the solution component along the dominating direction of the data matrix, the second term is the component along the second most dominating direction, and so forth.[8]

Thus, if we prefer to use the ordering induced by the SVD, then we should choose the matrix Z_k in (7.3) equal to the first k right singular vectors of A (we assume that $k \leq r$):

$$Z_k = V_k = \begin{pmatrix} v_1 & v_2 & \dots & v_k \end{pmatrix}.$$

Using the fact that

$$V^T V_k = \begin{pmatrix} I_k \\ 0 \end{pmatrix},$$

[8] However, this does not mean that the terms in the sum are ordered by magnitude.

where $I_k \in \mathbb{R}^{k\times k}$, we get

$$\|AV_k y - b\|_2^2 = \|U\Sigma V^T V_k y - b\|_2^2 = \left\| U\left(\Sigma \begin{pmatrix} I_k \\ 0 \end{pmatrix} y - U^T b\right)\right\|_2^2$$
$$= \left\| \begin{pmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_k \end{pmatrix} \begin{pmatrix} y_1 \\ \vdots \\ y_k \end{pmatrix} - \begin{pmatrix} u_1^T b \\ \vdots \\ u_k^T b \end{pmatrix} \right\|_2^2 + \sum_{i=k+1}^{r} (u_i^T b)^2.$$

We see that the least squares problem $\min_y \|AV_k y - b\|_2$ has the solution

$$y = \begin{pmatrix} u_1^T b/\sigma_1 \\ \vdots \\ u_k^T b/\sigma_k \end{pmatrix},$$

which is equivalent to taking

$$x_k := \sum_{i=1}^{k} \frac{u_i^T b}{\sigma_i} v_i$$

as an approximate solution of (7.1). This is often referred to as the *truncated SVD solution*.

Often one wants to find as low a value of k such that the reduction of the residual is substantial enough. The procedure can be formulated as an algorithm, which is sometimes referred to as *principal component regression*.

Principal component regression (truncated SVD)

1. Find the smallest value of k such that $\sum_{i=k+1}^{r} (u_i^T b)^2 < \mathrm{tol}\, \|b\|_2^2$.
2. Put

$$x_k := \sum_{i=1}^{k} \frac{u_i^T b}{\sigma_i} v_i.$$

The parameter tol is a predefined tolerance.

Example 7.2. We use the matrix from Example 1.1. Let

$$A = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 \end{pmatrix}.$$

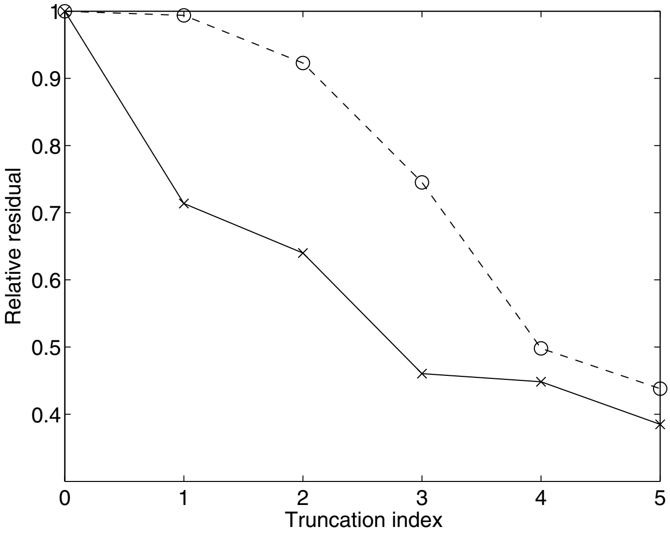


Figure 7.1. *The relative norm of the residuals for the query vectors* q_1 (*solid line*) *and* q_2 (*dashed*) *as functions of the truncation index* k.

Recall that each column corresponds to a document (here a sentence). We want to see how well two query vectors q_1 and q_2 can be represented in terms of the first few terms of the singular value expansion (7.4) of the solution, i.e., we will solve the least squares problems

$$\min_y \|AV_k y - q_i\|_2, \qquad i = 1, 2,$$

for different values of k. The two vectors are

$$q_1 = \begin{pmatrix} 0\\0\\0\\0\\0\\0\\0\\1\\1\\1 \end{pmatrix}, \qquad q_2 = \begin{pmatrix} 0\\1\\1\\0\\0\\0\\0\\0\\0\\0 \end{pmatrix},$$

corresponding to the words *rank*, *page*, *Web* and *England*, *FIFA*, respectively. In Figure 7.1 we plot the relative norm of residuals, $\|Ax_k - b\|_2/\|b\|_2$, for the two vectors as functions of k. From Example 1.1 we see that the main contents of the documents are related to the ranking of Web pages using the Google matrix, and this is reflected in the dominant singular vectors.[9] Since q_1 "contains Google

[9] See also Example 11.8 in Chapter 11.

terms," it can be well represented in terms of the first few singular vectors. On the other hand, the q_2 terms are related only to the "football document." Therefore, it is to be expected that the residual for q_1 decays faster than that of q_2 as a function of k.

The coordinates of q_1 and q_2 in terms of the first five left singular vectors are

```
U'*[q1 q2] = 1.2132    0.1574
            -0.5474    0.5215
             0.7698    0.7698
            -0.1817    0.7839
             0.3981   -0.3352
```

The vector q_1 has a substantial component in the first left singular vector u_1, and therefore the residual is reduced substantially for $k = 1$. Since q_2 has a small component in terms of u_1, there is only a marginal reduction of the residual in the first step.

If we want to reduce the relative residual to under 0.7 in this example, then we should choose $k = 2$ for q_1 and $k = 4$ for q_2. ■

7.2 A Krylov Subspace Method

When we use the truncated SVD (principal component regression) for a reduced-rank model, the right-hand side does not influence the choice of basis vectors z_i at all. The effect of this is apparent in Example 7.2, where the rate of decay of the residual is considerably slower for the vector q_2 than for q_1.

In many situations one would like to have a fast decay of the residual as a function of the number of basis vectors for any right-hand side. Then it is necessary to let the right-hand side influence the choice of basis vectors. This is done in an algorithm called *Lanczos–Golub–Kahan* (*LGK*) *bidiagonalization*, in the field of numerical linear algebra.[10] A closely related method is known in chemometrics and other areas as *partial least squares* or *projection to latent structures* (PLS). It is an algorithm out of a large class of *Krylov subspace methods*, often used for the solution of sparse linear systems; see, e.g., [42, Chapters 9–10], [80] or, for eigenvalue–singular value computations, see Section 15.8.

Krylov subspace methods are recursive, but in our derivation we will start with the reduction of a matrix to bidiagonal form using Householder transformations. The presentation in this section is largely influenced by [15].

7.2.1 Bidiagonalization Using Householder Transformations

The first step in the algorithm for computing the SVD of a dense matrix[11] $C \in \mathbb{R}^{m\times(n+1)}$ is to reduce it to upper bidiagonal form by Householder transformations

[10] The algorithm is often called Lanczos bidiagonalization, but it was first described by Golub and Kahan in [41].

[11] We choose these particular dimensions here because later in this chapter we will have $C = (b \;\; A)$.

from the left and right. We assume that $m > n$. The result is

$$C = P \begin{pmatrix} \hat{B} \\ 0 \end{pmatrix} W^T, \tag{7.5}$$

where P and W are orthogonal and $\hat{B}$ is upper bidiagonal. The decomposition in itself is useful also for other purposes. For instance, it is often used for the approximate solution of least squares problems, both dense and sparse.

We illustrate the Householder bidiagonalization procedure with a small example, where $C \in \mathbb{R}^{6\times 5}$. First, all subdiagonal elements in the first column are zeroed by a transformation P_1^T from the left (the elements that are changed in the transformation are denoted by $*$):

$$P_1^T C = P_1^T \begin{pmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \end{pmatrix} = \begin{pmatrix} * & * & * & * & * \\ 0 & * & * & * & * \\ 0 & * & * & * & * \\ 0 & * & * & * & * \\ 0 & * & * & * & * \\ 0 & * & * & * & * \end{pmatrix}.$$

Then, by a different Householder transformation W_1 from the right, we zero elements in the first row, from position 3 to n. To achieve this we choose

$$\mathbb{R}^{5\times 5} \ni W_1 = \begin{pmatrix} 1 & 0 \\ 0 & Z_1 \end{pmatrix},$$

where Z_1 is a Householder transformation. Since this transformation does not change the elements in the first column, the zeros that we just introduced in the first column remain. The result of the first step is

$$P_1^T C W_1 = \begin{pmatrix} \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \end{pmatrix} W_1 = \begin{pmatrix} \times & * & 0 & 0 & 0 \\ 0 & * & * & * & * \\ 0 & * & * & * & * \\ 0 & * & * & * & * \\ 0 & * & * & * & * \\ 0 & * & * & * & * \end{pmatrix} =: C_1.$$

We now continue in an analogous way and zero all elements below the diagonal in the second column by a transformation from the left. The matrix P_2 is constructed so that it does not change the elements in the first row of C_1, i.e., P_2 has the structure

$$\mathbb{R}^{6\times 6} \ni P_2 = \begin{pmatrix} 1 & 0 \\ 0 & \tilde{P}_2 \end{pmatrix},$$

where $\tilde{P}_2 \in \mathbb{R}^{5\times 5}$ is a Householder transformation. We get

$$P_2^T C_1 = \begin{pmatrix} \times & \times & 0 & 0 & 0 \\ 0 & * & * & * & * \\ 0 & 0 & * & * & * \\ 0 & 0 & * & * & * \\ 0 & 0 & * & * & * \\ 0 & 0 & * & * & * \end{pmatrix}.$$

Then, by a transformation from the right,

$$W_2 = \begin{pmatrix} I_2 & 0 \\ 0 & Z_2 \end{pmatrix}, \qquad I_2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix},$$

we annihilate elements in the second row without destroying the newly introduced zeros:

$$P_1^T C_1 W_2 = \begin{pmatrix} \times & \times & 0 & 0 & 0 \\ 0 & \times & * & 0 & 0 \\ 0 & 0 & * & * & * \\ 0 & 0 & * & * & * \\ 0 & 0 & * & * & * \\ 0 & 0 & * & * & * \end{pmatrix} =: C_2.$$

We continue in an analogous manner and finally obtain

$$P^T C W = \begin{pmatrix} \times & \times & & & \\ & \times & \times & & \\ & & \times & \times & \\ & & & \times & \times \\ & & & & \times \\ & & & & \end{pmatrix} = \begin{pmatrix} \hat{B} \\ 0 \end{pmatrix}. \tag{7.6}$$

In the general case,

$$P = P_1 P_2 \cdots P_n \in \mathbb{R}^{m\times m}, \qquad W = W_1 W_2 \cdots W_{n-2} \in \mathbb{R}^{(n+1)\times(n+1)}$$

are products of Householder transformations, and

$$\hat{B} = \begin{pmatrix} \beta_1 & \alpha_1 & & & \\ & \beta_2 & \alpha_2 & & \\ & & \ddots & \ddots & \\ & & & \beta_n & \alpha_n \\ & & & & \beta_{n+1} \end{pmatrix} \in \mathbb{R}^{(n+1)\times(n+1)}$$

is upper bidiagonal.

Due to the way the orthogonal matrices were constructed, they have a particular structure that will be used in the rest of this chapter.

Proposition 7.3. *Denote the columns of P in the bidiagonal decomposition* (7.6) *by p_i, $i = 1, 2, \dots, m$. Then*

$$p_1 = \beta_1 c_1, \qquad W = \begin{pmatrix} 1 & 0 \\ 0 & Z \end{pmatrix},$$

where c_1 is the first column of C and $Z \in \mathbb{R}^{n\times n}$ is orthogonal.

Proof. The first relation follows immediately from $P^T c_1 = \beta_1 e_1$. The second follows from the fact that all W_i have the structure

$$W_i = \begin{pmatrix} I_i & 0 \\ 0 & Z_i \end{pmatrix},$$

where $I_i \in \mathbb{R}^{i\times i}$ are identity matrices and Z_i are orthogonal. □

The reduction to bidiagonal form by Householder transformation requires $4mn^2 - 4n^3/3$ flops. If $m \gg n$, then it is more efficient to first reduce A to upper triangular form and then bidiagonalize the R factor.

Assume now that we want to solve the least squares problem $\min_x \|b - Ax\|_2$, where $A \in \mathbb{R}^{m\times n}$. If we choose $C = \begin{pmatrix} b & A \end{pmatrix}$ in the bidiagonalization procedure, then we get an equivalent bidiagonal least squares problem. Using (7.6) and Proposition 7.3 we obtain

$$P^T C W = P^T \begin{pmatrix} b & A \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & Z \end{pmatrix} = \begin{pmatrix} P^T b & P^T A Z \end{pmatrix} = \begin{pmatrix} \beta_1 e_1 & B \\ 0 & 0 \end{pmatrix}, \tag{7.7}$$

where

$$B = \begin{pmatrix} \alpha_1 & & & \\ \beta_2 & \alpha_2 & & \\ & \ddots & \ddots & \\ & & \beta_n & \alpha_n \\ & & & \beta_{n+1} \end{pmatrix} \in \mathbb{R}^{(n+1)\times n}.$$

Then, defining $y = Z^T x$ we can write the norm of the residual,

$$\begin{aligned} \|b - Ax\|_2 &= \left\| \begin{pmatrix} b & A \end{pmatrix} \begin{pmatrix} 1 \\ -x \end{pmatrix} \right\|_2 = \left\| P^T \begin{pmatrix} b & A \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & Z \end{pmatrix} \begin{pmatrix} 1 \\ -y \end{pmatrix} \right\|_2 \\ &= \left\| \begin{pmatrix} P^T b & P^T A Z \end{pmatrix} \begin{pmatrix} 1 \\ -y \end{pmatrix} \right\|_2 = \|\beta_1 e_1 - By\|_2. \end{aligned} \tag{7.8}$$

The bidiagonal least squares problem $\min_y \|\beta_1 e_1 - By\|_2$ can be solved in $O(n)$ flops, if we reduce B to upper bidiagonal form using a sequence of plane rotations (see below).

7.2.2 LGK Bidiagonalization

We will now give an alternative description of the bidiagonalization procedure of the preceding section that allows us to compute the decomposition (7.7) in a recursive manner. This is the LGK bidiagonalization. Part of the last equation of (7.7) can be written

$$P^T A = \begin{pmatrix} BZ^T \\ 0 \end{pmatrix}, \qquad BZ^T \in \mathbb{R}^{(n+1)\times n},$$

which implies

$$\begin{aligned} &A^T \begin{pmatrix} p_1 & p_2 & \cdots & p_{n+1} \end{pmatrix} \\ &= ZB^T = \begin{pmatrix} z_1 & z_2 & \dots & z_n \end{pmatrix} \begin{pmatrix} \alpha_1 & \beta_2 & & & & & \\ & \alpha_2 & \beta_3 & & & & \\ & & \ddots & \ddots & & & \\ & & & \beta_i & & & \\ & & & \alpha_i & & & \\ & & & & \ddots & \ddots & \\ & & & & & \alpha_n & \beta_{n+1} \end{pmatrix}. \end{aligned}$$

Equating column i ($i \geq 2$) on both sides, we get

$$A^T p_i = \beta_i z_{i-1} + \alpha_i z_i,$$

which can be written

$$\alpha_i z_i = A^T p_i - \beta_i z_{i-1}. \tag{7.9}$$

Similarly, by equating column i in

$$\begin{aligned} AZ &= A \begin{pmatrix} z_1 & z_2 & \dots & z_n \end{pmatrix} \\ &= PB = \begin{pmatrix} p_1 & p_2 & \dots & p_{n+1} \end{pmatrix} \begin{pmatrix} \alpha_1 & & & & & \\ \beta_2 & \alpha_2 & & & & \\ & \ddots & \ddots & & & \\ & & & \alpha_i & & \\ & & & \beta_{i+1} & & \\ & & & & \ddots & \ddots \\ & & & & \beta_n & \alpha_n \\ & & & & & \beta_{n+1} \end{pmatrix}, \end{aligned}$$

we get

$$Az_i = \alpha_i p_i + \beta_{i+1} p_{i+1},$$

which can be written

$$\beta_{i+1} p_{i+1} = Az_i - \alpha_i p_i. \tag{7.10}$$

Now, by compiling the starting equation $\beta_1 p_1 = b$ from Proposition 7.3, equations (7.9) and (7.10), we have derived a recursion:

LGK Bidiagonalization

```
1. $\beta_1 p_1 = b$, $z_0 = 0$
2. for $i = 1 : n$
       $\alpha_i z_i = A^T p_i - \beta_i z_{i-1}$,
       $\beta_{i+1} p_{i+1} = A z_i - \alpha_i p_i$
3. end
```

The coefficients α_{i-1} and β_i are determined so that $\|p_i\| = \|z_i\| = 1$.

The recursion breaks down if any α_i or β_i becomes equal to zero. It can be shown (see, e.g., [15, Section 7.2]) that in the solution of least squares problems, these occurrences are harmless in the sense that they correspond to well-defined special cases.

The recursive bidiagonalization procedure gives, *in exact arithmetic*, the same result as the Householder bidiagonalization of $\begin{pmatrix} b & A \end{pmatrix}$, and thus the generated vectors $(p_i)_{i=1}^n$ and $(z_i)_{i=1}^n$ satisfy $p_i^T p_j = 0$ and $z_i^T z_j = 0$ if $i \neq j$. However, in floating point arithmetic, the vectors lose orthogonality as the recursion proceeds; see Section 7.2.7.

7.2.3 Approximate Solution of a Least Squares Problem

Define the matrices $P_k = \begin{pmatrix} p_1 & p_2 & \dots & p_k \end{pmatrix}$, $Z_k = \begin{pmatrix} z_1 & z_2 & \dots & z_k \end{pmatrix}$, and

$$B_k = \begin{pmatrix} \alpha_1 & & & \\ \beta_2 & \alpha_2 & & \\ & \ddots & \ddots & \\ & & \beta_{k-1} & \alpha_{k-1} \\ & & & \beta_k \end{pmatrix} \in \mathbb{R}^{k \times (k-1)}.$$

In the same way we could write the relations $AZ = PB$ and $A^T P = ZB^T$ as a recursion, we can now write the *first k steps of the recursion* as a matrix equation

$$AZ_k = P_{k+1} B_{k+1}. \qquad (7.11)$$

Consider the least squares problem $\min_x \|Ax - b\|_2$. Note that the column vectors z_i are orthogonal vectors in $\mathbb{R}^n$, where the solution x lives. Assume that we want to *find the best approximate solution in the subspace spanned by the vectors $z_1, z_2, \dots, z_k$*. That is equivalent to solving the least squares problem

$$\min_y \|AZ_k y - b\|_2,$$

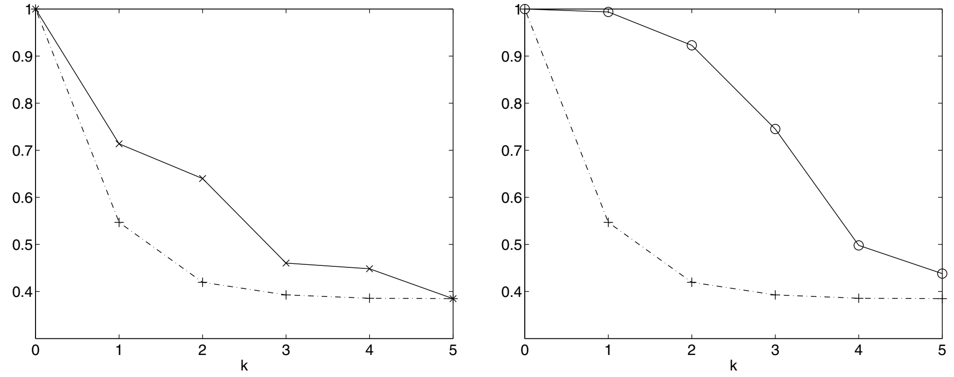


Figure 7.2. *The relative norm of the residuals for the query vectors* q_1 *(left) and* q_2 *(right) as a function of subspace dimension* k*. The residual curves for the truncated SVD solutions are solid and for the bidiagonalization solutions are dash-dotted.*

where $y \in \mathbb{R}^k$. From (7.11) we see that this is the same as solving

$$\min_y \|P_{k+1}B_{k+1}y - b\|_2,$$

which, using the orthogonality of $P = \begin{pmatrix} P_{k+1} & P_\perp \end{pmatrix}$, we can rewrite

$$\begin{aligned}\|P_{k+1}B_{k+1}y - b\|_2 &= \|P^T(P_{k+1}B_{k+1}y - b)\|_2 \\ &= \left\| \begin{pmatrix} P_{k+1}^T \\ P_\perp^T \end{pmatrix} (P_{k+1}B_{k+1}y - b) \right\|_2 = \left\| \begin{pmatrix} B_{k+1}y \\ 0 \end{pmatrix} - \begin{pmatrix} \beta_1 e_1 \\ 0 \end{pmatrix} \right\|_2,\end{aligned}$$

since $b = \beta_1 p_1$. It follows that

$$\min_y \|AZ_k y - b\|_2 = \min_y \|B_{k+1}y - \beta_1 e_1\|_2, \qquad (7.12)$$

which, due to the bidiagonal structure, we can solve in $O(n)$ flops; see below.

Example 7.4. Using bidiagonalization, we compute approximate solutions to the same least squares problems as in Example 7.2. The relative norm of the residual, $\|AZ_k y - b\|_2/\|b\|_2$, is plotted as a function of k in Figure 7.2 for the truncated SVD solution and the bidiagonalization procedure. It is seen that in both cases, the bidiagonalization-based method give a faster decay of the residual than the truncated SVD solutions. Thus in this example, the fact that we let the basis vectors z_i be influenced by the right-hand sides q_1 and q_2 leads to reduced rank models of smaller dimensions. If we want to reduce the relative residual to below 0.7, then in both cases we can choose $k = 1$ with the bidiagonalization method. ■

The least squares problem (7.12) with bidiagonal structure can be solved using a sequence of plane rotations. Consider the reduction of

$$\begin{pmatrix} B_{k+1} & \beta e_1 \end{pmatrix} = \begin{pmatrix} \alpha_1 & & & & & \beta_1 \\ \beta_2 & \alpha_2 & & & & 0 \\ & \beta_3 & \alpha_3 & & & 0 \\ & & \ddots & \ddots & & \vdots \\ & & & \beta_k & \alpha_k & 0 \\ & & & & \beta_{k+1} & 0 \end{pmatrix}$$

to upper triangular form. We will now demonstrate that the norm of the residual can be easily computed. In the first step we zero β_2 by a rotation in the $(1, 2)$ plane, with cosine and sine c_1 and s_1. The result is

$$\begin{pmatrix} \hat{\alpha}_1 & + & & & & \beta_1 \\ 0 & \hat{\alpha}_2 & & & & -\beta_1 s_1 \\ & \beta_3 & \alpha_3 & & & 0 \\ & & \ddots & \ddots & & \vdots \\ & & & \beta_k & \alpha_k & 0 \\ & & & & \beta_{k+1} & 0 \end{pmatrix},$$

where matrix elements that have changed are marked with a hat, and the new nonzero element is marked with a $+$. In the next step, we zero β_3 by a rotation with cosine and sine c_2 and s_2:

$$\begin{pmatrix} \hat{\alpha}_1 & + & & & & \beta_1 \\ 0 & \hat{\alpha}_2 & + & & & -\beta_1 s_1 \\ & 0 & \hat{\alpha}_3 & & & \beta_1 s_1 s_2 \\ & & \ddots & \ddots & & \vdots \\ & & & \beta_k & \alpha_k & 0 \\ & & & & \beta_{k+1} & 0 \end{pmatrix}.$$

The final result after k steps is

$$\begin{pmatrix} \hat{\alpha}_1 & + & & & & \gamma_0 \\ & \hat{\alpha}_2 & + & & & \gamma_1 \\ & & \hat{\alpha}_3 & + & & \gamma_2 \\ & & & \ddots & & \vdots \\ & & & & \hat{\alpha}_k & \gamma_{k-1} \\ & & & & & \gamma_k \end{pmatrix} =: \begin{pmatrix} \widehat{B}_k & \gamma \\ 0 & \gamma_k \end{pmatrix},$$

where $\gamma_i = (-1)^i \beta_1 s_1 s_2 \cdots s_i$ and $\gamma^{(k)} = \begin{pmatrix} \gamma_0 & \gamma_1 & \cdots & \gamma_{k-1} \end{pmatrix}^T$. If we define the product of plane rotations to be the orthogonal matrix $Q_{k+1} \in \mathbb{R}^{(k+1)\times(k+1)}$, we have the QR decomposition

$$B_{k+1} = Q_{k+1} \begin{pmatrix} \widehat{B}_k \\ 0 \end{pmatrix} \tag{7.13}$$

and

$$\begin{pmatrix} \gamma^{(k)} \\ \gamma_k \end{pmatrix} = \begin{pmatrix} \gamma_0 \\ \gamma_1 \\ \vdots \\ \gamma_k \end{pmatrix} = Q_{k+1}^T \begin{pmatrix} \beta_1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}. \tag{7.14}$$

Using the QR decomposition we can write

$$\|B_{k+1}y - \beta_1 e_1\|_2^2 = \|\widehat{B}_k y - \gamma\|_2^2 + |\gamma_k|^2,$$

and the norm of the residual in the least squares problem is equal to $|\gamma_k| = |\beta_1 s_1 \cdots s_k|$. It follows that the norm of the residual can be computed recursively as we generate the scalar coefficients α_i and β_i, and thus it is possible to monitor the decay of the residual.

7.2.4 Matrix Approximation

The bidiagonalization procedure also gives a low-rank approximation of the matrix A. Here it is slightly more convenient to consider the matrix A^T for the derivation. Assume that we want to use the columns of Z_k as approximate basis vectors in $\mathbb{R}^n$. Then we can determine the coordinates of the columns of A^T in terms of this basis by solving the least squares problem

$$\min_{S_k \in \mathbb{R}^{m\times k}} \|A^T - Z_k S_k^T\|_F. \tag{7.15}$$

Lemma 7.5. *Given the matrix $A \in \mathbb{R}^{m\times n}$ and the matrix $Z_k \in \mathbb{R}^{n\times k}$ with orthonormal columns, the least squares problem* (7.15) *has the solution*

$$S_k = P_{k+1} B_{k+1}.$$

Proof. Since the columns of Z_k are orthonormal, the least squares problem has the solution

$$S_k^T = Z_k^T A^T,$$

which by (7.11) is the same as $S_k = P_{k+1}B_{k+1}$. □

From the lemma we see that we have a least squares approximation $A^T \approx Z_k(P_{k+1}B_{k+1})^T$ or, equivalently,

$$A \approx P_{k+1}B_{k+1}Z_k^T.$$

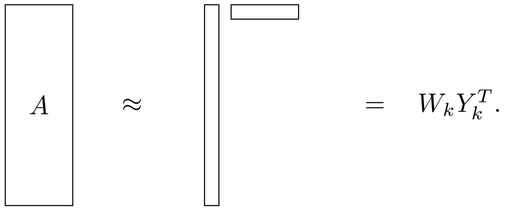
However, this is not a "proper" rank-k approximation, since $P_{k+1} \in \mathbb{R}^{m\times(k+1)}$ and $B_{k+1} \in \mathbb{R}^{(k+1)\times k}$. Now, with the QR decomposition (7.13) of B_{k+1} we have

$$P_{k+1}B_{k+1} = (P_{k+1}Q_{k+1})(Q_{k+1}^T B_{k+1}) = (P_{k+1}Q_{k+1}) \begin{pmatrix} \widehat{B}_k \\ 0 \end{pmatrix} = W_k \widehat{B}_k,$$

where W_k is defined to be the first k columns of $P_{k+1}Q_{k+1}$. With $Y_k^T = \widehat{B}_k Z_k^T$ we now have a proper rank-k approximation of A:

$$A \approx P_{k+1}B_{k+1}Z_k^T = W_k Y_k^T, \qquad W_k \in \mathbb{R}^{m\times k}, \qquad Y_k \in \mathbb{R}^{n\times k}. \tag{7.16}$$

The low-rank approximation of A is illustrated as



As before, we can interpret the low-rank approximation as follows. *The columns of W_k are a basis in a subspace of R^m. The coordinates of column j of A in this basis are given in column j of Y_k^T.*

7.2.5 Krylov Subspaces

In the LGK bidiagonalization, we create two sets of basis vectors—the p_i and the z_i. It remains to demonstrate what subspaces they span. From the recursion we see that z_1 is a multiple of A^Tb and that p_2 is a linear combination of b and AA^Tb. By an easy induction proof one can show that

$$p_k \in \operatorname{span}\{b,\ AA^Tb,\ (AA^T)^2b,\ \ldots,\ (AA^T)^{k-1}b\},$$
$$z_k \in \operatorname{span}\{A^Tb,\ (A^TA)A^Tb,\ \ldots,\ (A^TA)^{k-1}A^Tb\}$$

for $k = 1, 2, \ldots$. Denote

$$\mathcal{K}_k(C, b) = \operatorname{span}\{b,\ Cb,\ C^2,\ \ldots,\ C^{k-1}b\}.$$

This a called a *Krylov subspace*. We have the following result.

Proposition 7.6. *The columns of P_k are an orthonormal basis of $\mathcal{K}_k(AA^T, b)$, and the columns of Z_k are an orthonormal basis of $\mathcal{K}_k(A^TA, A^Tb)$.*

7.2.6 Partial Least Squares

Partial least squares (PLS) [109, 111] is a recursive algorithm for computing approximate least squares solutions and is often used in chemometrics. Different variants of the algorithm exist, of which perhaps the most common is the so-called NIPALS formulation.

The NIPALS PLS algorithm

```
1. $A_0 = A$
2. for i=1,2,...,k
    (a) $w_i = \frac{1}{\|A_{i-1}^T b\|} A_{i-1}^T b$
    (b) $\tilde{u}_i = \frac{1}{\|A_{i-1} w_i\|} A_{i-1} w_i$
    (c) $\tilde{v}_i = A_{i-1}^T \tilde{u}_i$
    (d) $A_i = A_{i-1} - \tilde{u}_i \tilde{v}_i^T$
```

This algorithm differs from LGK bidiagonalization in a few significant ways, the most important being that the data matrix is deflated as soon as a new pair of vectors $(\tilde{u}_i, \tilde{v}_i)$ has been computed. However, it turns out [32, 110] that the PLS algorithm is mathematically equivalent to a variant of LGK bidiagonalization that is started by choosing not p_1 but instead $\alpha_1 z_1 = A^T b$. This implies that the vectors $(w_i)_{i=1}^k$ form an orthonormal basis in $\mathcal{K}_k(A^T A, A^T b)$, and $(\tilde{u}_i)_{i=1}^k$ form an orthonormal basis in $\mathcal{K}_k(AA^T, AA^T b)$.

7.2.7 Computing the Bidiagonalization

The recursive versions of the bidiagonalization suffers from the weakness that the generated vectors lose orthogonality. This can be remedied by *reorthogonalizing* the vectors, using a Gram–Schmidt process. Householder bidiagonalization, on the other hand, generates vectors that are as orthogonal as can be expected in floating point arithmetic; cf. Section 4.4. Therefore, for dense matrices A of moderate dimensions, one should use this variant.[12]

For large and sparse or otherwise structured matrices, it is usually necessary to use the recursive variant. This is because the Householder algorithm modifies the matrix by orthogonal transformations and thus destroys the structure. Note that for such problems, the PLS algorithm has the same disadvantage because it deflates the matrix (step (d) in the algorithm above).

A version of LGK bidiagonalization that avoids storing all the vectors p_i and z_i has been developed [75].

[12] However, if there are missing entries in the matrix, which is often the case in certain applications, then the PLS algorithm can be modified to estimate those; see, e.g., [111].