Journal of VLSI Signal Processing, 8, 1–9 (1996)

A Symmetric Rank-Revealing Toeplitz Matrix Decomposition

FRANKLIN T. LUK
Computer Science Department, Rensselaer Polytechnic Institute, Troy, New York 12180, USA

SANZHENG QIAO
Communications Research Laboratory, McMaster University, Hamilton, Ontario L8S 4K1, Canada

Abstract. In signal and image processing, regularization often requires a rank-revealing decomposition of a symmetric Toeplitz matrix with a small rank deficiency. In this paper, we present an efficient factorization method that exploits symmetry as well as the rank and Toeplitz properties of the given matrix.

Keywords: Toeplitz matrix, regularization, symmetric rank-revealing decomposition

1. Introduction

In signal and image processing applications [5], [6], a noisy and distorted signal vector $\hat{x}$ is given by

$$\hat{x} = Tx + w, \tag{1}$$

where x and w represent an unknown original signal vector and a noise vector, respectively, and T is a predetermined matrix describing the spread of signals. This problem arises often in array processing, where the matrix T may be real, symmetric, and Toeplitz. Assuming the dimensions of T to be $n \times n$, we have

$$T = \begin{pmatrix} t_1 & t_2 & t_3 & \dots & t_n \\ t_2 & t_1 & t_2 & \dots & t_{n-1} \\ t_3 & t_2 & t_1 & \dots & t_{n-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ t_n & t_{n-1} & t_{n-2} & \dots & t_1 \end{pmatrix}.$$

To restore the original signal vector x from $\hat{x}$, we should invert T. Quite frequently, cf. [1], T is ill-conditioned and regularization is required. One popular method is the truncated singular value decomposition (TSVD) [4], which requires $O(n^3)$ floating-point operations (flops) because it is not known how to compute the singular value decomposition (SVD) of a Toeplitz matrix in fewer flops.

For general matrices, less expensive rank-revealing methods like the URV decomposition have been developed by Stewart [7] and others to replace the SVD. But $O(n^3)$ flops are still required. If the matrix is Toeplitz and banded with bandwidth b, Nagy [6] proposed an $O(bn^2)$ method for computing an approximate URV decomposition. However, no one has shown how to exploit the symmetry of T. We suggest a possible approach in Section 2.

In addition, assume that T has a small rank deficiency, viz.,

$$\text{rank}(T) = n - k,$$

where k denotes a small integer. We will present an $O(kn^2)$ method in Sections 3 to 5 for computing a rank-revealing factorization. Our other contribution is to show how to avoid complex arithmetic in the computation. The examples in Section 6 show that our new method restores the signal almost as accurately as the TSVD approach.

The paper is organized as follows. We present an extension of rank-revealing factorizations to symmetric matrices in Section 2. In Section 3 we sketch a fast $O(n^2)$ triangularization scheme based on hyperbolic rotations, and in Section 4 we show how to avoid the use of complex arithmetic when these rotations are applied. Lastly, in Sections 5 and 6, we discuss rank-revealing techniques and present simulation results.

2. Use Symmetry

Many popular regularization techniques are based on an SVD of T:

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

where U and V are orthogonal, and $\Sigma = \text{diag}(\sigma_i)$, with

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

Select a tolerance τ to find k such that

$$\sigma_{n-k} > \tau \geq \sigma_{n-k+1}.$$

We may say that T has k small singular values, or that T has a numerical rank of $n-k$. Let

$$\hat{\Sigma} = \text{diag}(\sigma_1, \ldots, \sigma_{n-k}, 0, \ldots, 0).$$

We restore the original signal x via

$$x = V\hat{\Sigma}^+ U^T \hat{x},$$

where $\hat{\Sigma}^+ = \text{diag}(\sigma_i^+)$ denotes the pseudo-inverse of $\hat{\Sigma}$.

Although the TSVD is stable, it is expensive to compute. Since regularization does not require the diagonalization of T, we may pick the less costly URV decomposition:

$$T = URV^T,$$

where U and V are orthogonal, and R is upper triangular, viz.,

$$R = \begin{pmatrix} \bar{R} & E \\ 0 & G \end{pmatrix}.$$

The three submatrices $\bar{R}$, E and G ($k \times k$) possess the special properties that $\|\bar{R}\|$ is large, and $\|E\|$ and $\|G\|$ are small:

$$\begin{cases} \sigma_{\min}(\bar{R}) \approx \sigma_{n-k}\,, \\ \\ \|E\|_F^2 + \|G\|_F^2 \approx \sigma_{n-k+1}^2 + \cdots + \sigma_n^2\,. \end{cases} \tag{2}$$

A regularized solution to (1) is obtained from

$$x = V \begin{pmatrix} \bar{R}^{-1} & 0 \\ 0 & 0 \end{pmatrix} U^T \hat{x}.$$

We wish to exploit the symmetry of T to save on storage and work. Start by computing a symmetric eigenvalue decomposition:

$$T = V\Lambda V^T,$$

where V is orthogonal, and $\Lambda = \text{diag}(\lambda_i)$, with

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

Choose a tolerance τ to find k such that

$$|\lambda_{n-k}| > \tau \geq |\lambda_{n-k+1}|,$$

i.e., T has a numerical rank of $n-k$. Let

$$\hat{\Lambda} = \text{diag}(\lambda_1, \cdots, \lambda_{n-k}, 0, \cdots, 0).$$

The original signal is restored by the formula:

$$x = V\hat{\Lambda}^+ V^T \hat{x}.$$

where $\hat{\Lambda}^+ = \text{diag}(\lambda^+)$.

We propose to generalize the URV decomposition as follows:

$$T = VSV^T, \tag{3}$$

where V is orthogonal and S symmetric. We shall call (3) a VSV decomposition. Partition S:

$$S \equiv \begin{pmatrix} \bar{S} & E \\ E^T & G \end{pmatrix}, \tag{4}$$

where the three submatrices $\bar{S}$, E and G ($k \times k$) possess similar norm properties as in (2):

$$\begin{cases} \lambda_{\min}(\bar{S}) \approx \lambda_{n-k}\,, \\ \\ \|E\|_F^2 + \|\text{triu}(G)\|_F^2 \approx \lambda_{n-k+1}^2 + \cdots + \lambda_n^2\,. \end{cases} \tag{5}$$

We use $\text{triu}(G)$ to denote the upper triangular part of G. The signal vector x can be restored by a truncated VSV decomposition:

$$x = V \begin{pmatrix} \bar{S}^{-1} & 0 \\ 0 & 0 \end{pmatrix} V^T \hat{x}.$$

Note that (5) is not an obvious extension of (2) to the symmetric case. Why do we use $\text{triu}(G)$? Roughly speaking, since E, E^T and G in (4) are small and G is symmetric, it is reasonable to exclude the redundant data and consider only E and $\text{triu}(G)$. A more rigorous argument will be given at the end of Section 5.

3. Fast Triangular Factorization

In this section we discuss a fast $O(n^2)$ triangularization of an $n \times n$ symmetric Toeplitz matrix.

First, assume that $t_1 \neq 0$ and $t_1 > 0$; otherwise consider $-T$. Use a displacement representation for T [2]:

$$T = R_1^T R_1 - R_2^T R_2, \tag{6}$$

where R_1 and R_2 are Toeplitz matrices:

$$R_1 = \frac{1}{\sqrt{t_1}} \begin{pmatrix} t_1 & t_2 & t_3 & \dots & t_{n-1} & t_n \\ 0 & t_1 & t_2 & \dots & t_{n-2} & t_{n-1} \\ 0 & 0 & t_1 & \dots & t_{n-3} & t_{n-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & t_1 & t_2 \\ 0 & 0 & 0 & \dots & 0 & t_1 \end{pmatrix}$$

and

$$R_2 = \frac{1}{\sqrt{t_1}} \begin{pmatrix} 0 & t_2 & t_3 & \dots & t_{n-1} & t_n \\ 0 & 0 & t_2 & \dots & t_{n-2} & t_{n-1} \\ 0 & 0 & 0 & \dots & t_{n-3} & t_{n-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 0 & t_2 \\ 0 & 0 & 0 & \dots & 0 & 0 \end{pmatrix}.$$

Rewrite (6) to get

$$T = (R_1^T \; R_2^T) \begin{pmatrix} I & 0 \\ 0 & -I \end{pmatrix} \begin{pmatrix} R_1 \\ R_2 \end{pmatrix},$$

and apply 2×2 real transformations (details in Section 4) to eliminate R_2:

$$T = (R^T \; 0) \begin{pmatrix} D_1 & 0 \\ 0 & D_2 \end{pmatrix} \begin{pmatrix} R \\ 0 \end{pmatrix} = R^T D R, \tag{7}$$

where R is upper triangular, and D_1 and D_2 are signature matrices (diagonal matrices with ± 1 on the diagonal). We take care to eliminate R_2 in a special order, so as to maintain the Toeplitz structures of R_1 and R_2. Rotating the second row of R_1 against the first row of R_2, we zero out the (1,2)-entry of R_2. But a whole super-diagonal of R_2 can be annihilated by applying this same rotation to the $(i+1)$-st row of R_1 and the i-th row of R_2, for $i = 1, \dots, n-1$. The key is that we store and operate upon R_1 and R_2 as vectors, in view of their Toeplitz structures. The calculation of the RDR decomposition defined by (7) requires only $O(n^2)$ flops.

It is well known that the RDR decomposition without pivoting is numerically unstable. Let

$$T = \begin{pmatrix} 1.00 & 0.999 & -0.602 \\ 0.999 & 1.00 & 0.999 \\ -0.602 & 0.999 & 1.00 \end{pmatrix}.$$

Using three-decimal-digit arithmetic with rounding, the procedure computes $D = \mathrm{diag}(1, 1, -1)$,

$$R = \begin{pmatrix} 1.00 & 0.999 & -0.602 \\ 0 & 0.0447 & 35.8 \\ 0 & 0 & 35.8 \end{pmatrix},$$

and

$$T = R^T D R + \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0.638 \end{pmatrix}.$$

The matrix T has a small condition number of 2.88; its eigenvalues are 2.14, 1.60, and -0.746. The problem is that its leading 2×2 principal submatrix is ill-conditioned. Pivoting can improve the stability, but it destroys the Toeplitz and symmetry structure, resulting in a slow algorithm. Since we are primarily interested in separating small eigenvalues from large ones, we can apply a moderately small shift to T to improve the numerical stability. For the above example, we may apply the procedure to a shifted T, e.g., $T + 0.1I$. Then

$$\hat{R} = \begin{pmatrix} 1.05 & 0.951 & -0.573 \\ 0 & 0.444 & 3.47 \\ 0 & 0 & 3.36 \end{pmatrix}$$

and

$$T + 0.1I = \hat{R}^T D \hat{R} + \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0.004 \\ 0 & 0.004 & 0.02 \end{pmatrix}.$$

So the factorization error is much smaller.

Second, when $t_1 = 0$, the RDR decomposition does not exist. We may use the above shifting technique to overcome the difficulty. However, the choice of an appropriate shift can be a delicate matter.

4. Avoid Complex Arithmetic

Since the given matrix T is real and the resultant matrices R and D are real, we want to restrict the computation to real arithmetic. In this section, we present details on how to construct a sequence of 2×2 real transformations Y to eliminate R_2.

Given two real quantities α and β, and two scalars $d_1 = \pm 1$ and $d_2 = \pm 1$, consider the problem of finding a real transformation Y so that

$$Y \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} \gamma \\ 0 \end{pmatrix} \tag{8}$$

and

$$Y^{-T} \begin{pmatrix} d_1 & 0 \\ 0 & d_2 \end{pmatrix} Y^{-1} = \begin{pmatrix} \hat{d}_1 & 0 \\ 0 & \hat{d}_2 \end{pmatrix}, \tag{9}$$

where $\hat{d}_1 = \pm 1$ and $\hat{d}_2 = \pm 1$. Then we get

$$(\alpha \quad \beta)\, Y^T Y^{-T} \begin{pmatrix} d_1 & 0 \\ 0 & d_2 \end{pmatrix} Y^{-1} Y \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = (\gamma \quad 0) \begin{pmatrix} \hat{d}_1 & 0 \\ 0 & \hat{d}_2 \end{pmatrix} \begin{pmatrix} \gamma \\ 0 \end{pmatrix}.$$

For simplicity, we will assume that

$$d_1 = 1\,.$$

If needed, we can factor out a minus sign to force $d_1 = 1$.

```
Q-Rotate ( α, β, d₂ )
    check that β ≠ 0 and |α| ≠ −d₂|β|;
    if |α| > |β| then
        h := 1 + d₂(β/α)²;
        r := √h · |α|;
    else
        h := (α/β)² + d₂;
        r := √|h| · |β|;
    c := α/r; s := β/r;
    if h > 0;
        γ := r; d̂₁ := 1; d̂₂ := d₂;
    else
        γ := −r; d̂₁ := −1; d̂₂ := −d₂;
    return( γ, d̂₁, d̂₂, c, s ).
End Q-Rotate
```

Fig. 1. Generate Y for relations (8) and (9).

To zero out β in (8), we choose

$$Z = \begin{pmatrix} c & d_2 s \\ -s & c \end{pmatrix}, \tag{10}$$

with $c = \alpha / r$ and $s = \beta / r$, where

$$r = \sqrt{\alpha^2 + d_2 \beta^2}.$$

So $c^2 + d_2 s^2 = 1$, and

$$Z^{-1} = \begin{pmatrix} c & -d_2 s \\ s & c \end{pmatrix}.$$

We see that (9) is satisfied:

$$Z^{-T} \begin{pmatrix} 1 & 0 \\ 0 & d_2 \end{pmatrix} Z^{-1} = \begin{pmatrix} 1 & 0 \\ 0 & d_2 \end{pmatrix}.$$

When $d_2 = 1$, we get a circular (also known as Givens) rotation Z, where

$$Z = \begin{pmatrix} c & s \\ -s & c \end{pmatrix}, \; Z^{-1} = Z^T,$$

and $c^2 + s^2 = 1$. When $d_2 = -1$, we get a hyperbolic rotation Z satisfying

$$Z = \begin{pmatrix} c & -s \\ -s & c \end{pmatrix}, \; Z^{-1} = \begin{pmatrix} c & s \\ s & c \end{pmatrix},$$

and $c^2 - s^2 = 1$. Thus, our new transformation Z unifies the circular and hyperbolic rotations. We refer to Z as a quadratic rotation.

When Z is real, viz., $\alpha^2 + d_2 \beta^2 > 0$, the choice is clear:

$$Y = Z.$$

But Z is complex when $d_2 = -1$ and $|\alpha| < |\beta|$, in which case we calculate

$$\tilde{r} = \sqrt{-(\alpha^2 + d_2 \beta^2)} > 0,$$

as well as $\tilde{c} = \alpha / \tilde{r}$ and $\tilde{s} = \beta / \tilde{r}$. Defining

$$Y = \begin{pmatrix} \tilde{c} & d_2 \tilde{s} \\ -\tilde{s} & \tilde{c} \end{pmatrix},$$

we see that Y is real and satisfies (8). So $\tilde{r} = ir$, where $i = \sqrt{-1}$. Thus,

$$Y = -iZ,$$

and $\hat{d}_1 = -1$, $\hat{d}_2 = -d_2$.

We summarize our results on quadratic rotations in a procedure called *Q-Rotate* in Figure 1.

5. Reveal Rank

Suppose that we have decomposed T into the triangular factorization of (7):

$$T = R^T D R.$$

In this section, we show how to use it to compute a rank-revealing factorization.

We start by finding a normalized vector z that approximates the eigenvector z_n corresponding to the smallest (in magnitude) eigenvalue λ_n. Then

$$z = z_n + u\,,$$

where u denotes an error vector satisfying

$$\|u\|_2 < \epsilon$$

for some small quantity ϵ. Use the technique in [7] to find circular rotations (call the product V^T) that transform z into the n-th unit vector e_n:

$$V^T z = e_n\,.$$

Apply the transpose of these rotations from the right on R. However, when a circular rotation is applied to the i-th and $(i+1)$-st columns of R, it creates a nonzero $(i+1, i)$ subdiagonal entry in R. To restore the triangular structure of R, apply a quadratic rotation from the left to annihilate the newly created nonzero element. We have

$$T = V S V^T,$$

where

$$S = \bar{R}^T \bar{D} \bar{R},$$

with $\bar{R}$ and $\bar{D}$ denoting, respectively, the resultant triangular and signature matrices. To reveal rank, we partition

$$S = \begin{pmatrix} \bar{S} & y \\ y^T & \lambda \end{pmatrix} \tag{11}$$

where $\bar{S}$ is $(n-1)\times(n-1)$; we also let

$$V = (V_1 \quad z),$$

where V_1 is $n\times(n-1)$. Now, we show that the matrix S satisfies the norm properties (5) with $k = 1$. First, since $\lambda = e_n^T S e_n$, we get

$$|\lambda_n| \le |\lambda|\,.$$

Also, since $S = V^T T V$ and $V e_n = z$, we have

$$\lambda = z^T T z = \lambda_n + 2\lambda_n u^T z_n + u^T T u\,,$$

and so

$$|\lambda| \le |\lambda_n| + 2|\lambda_n|\,\epsilon + \|T\|_2\,\epsilon^2.$$

Second, check the vector:

$$y = V_1^T T z = V_1^T T(z_n + u) = \lambda_n V_1^T z_n + V_1^T T u.$$

Since $V_1^T z = 0$, we have

$$V_1^T z_n = -V_1^T u\,.$$

It follows that

$$y = V_1^T (T - \lambda_n I) u$$

and

$$\|y\|_2 < \|V_1^T (T - \lambda_n I)\|_2\,\epsilon \le \|T - \lambda_n I\|_2\,\epsilon.$$

Consequently, when $z \approx z_n$, we get

$$\|y\|_2^2 + \lambda^2 \approx \lambda_n^2.$$

Third, check $\lambda_{\min}(\bar{S})$. Let

$$\bar{S} = \tilde{V} \tilde{D} \tilde{V}^T$$

be an eigenvalue decomposition of $\bar{S}$. Then

$$T = V S V^T = V \begin{pmatrix} \bar{S} & y \\ y^T & \lambda \end{pmatrix} V^T$$

and so

$$T = V \begin{pmatrix} \tilde{V} & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \tilde{D} & \tilde{V}^T y \\ y^T \tilde{V} & \lambda \end{pmatrix} \begin{pmatrix} \tilde{V}^T & 0 \\ 0 & 1 \end{pmatrix} V^T.$$

Gerschgorin theorem states that if ϵ is sufficiently small and if the eigenvalues of T are distinct, then

$$|\lambda_{\min}(\bar{S}) - \lambda_{n-1}| \le \|\tilde{V}^T y\|_2 \le \|T - \lambda_n I\|_2\,\epsilon\,,$$

showing that

$$\lambda_{\min}(\bar{S}) \approx \lambda_{n-1}.$$

In summary, if $z \approx z_n$ and the eigenvalues of T are distinct, then

$$\begin{cases} \|y\|_2^2 + \lambda^2 \approx \lambda_n^2\,, \\[2ex] \lambda_{\min}(\bar{S}) \approx \lambda_{n-1}\,, \end{cases}$$

which are simply the properties (5) when $k = 1$.

If the estimated smallest eigenvalue is less than the preset tolerance, we deflate T and repeat the procedure on $\bar{S}$ in (11). As before, we get

$$\bar{S} = \tilde{V} \begin{pmatrix} \tilde{S} & \tilde{y} \\ \tilde{y}^T & \tilde{\lambda} \end{pmatrix} \tilde{V}^T \tag{12}$$

and

$$\begin{cases} \|\tilde{y}\|_2^2 + \tilde{\lambda}^2 \approx \lambda_{n-1}^2 \,, \\ \lambda_{\min}(\tilde{S}) \approx \lambda_{n-2} \,. \end{cases}$$

Now,

$$\|y\|_2^2 + \|\tilde{y}\|_2^2 + \lambda^2 + \tilde{\lambda}^2 \approx \lambda_{n-1}^2 + \lambda_n^2.$$

Combining (11) and (12), we get

$$T = \hat{V} \begin{pmatrix} \tilde{S} & E \\ E^T & G \end{pmatrix} \hat{V}^T$$

where G is 2×2. Then we have

$$\begin{cases} \|E\|_F^2 + \|\mathrm{triu}(G)\|_F^2 \approx \lambda_{n-1}^2 + \lambda_n^2 \,, \\ \lambda_{\min}(\tilde{S}) \approx \lambda_{n-2} \,, \end{cases}$$

which are precisely (5) for the case where $k = 2$. This also justifies the use of $\mathrm{triu}(G)$ in (5).

We continue the deflation procedure until the estimated eigenvalue exceeds the tolerance. Consequently, we obtain the desired VSV decomposition of (3). This factorization, including the eigenvector estimation, costs $O(kn^2)$ flops.

Fig. 2. The 48 smallest eigenvalues in Example 3.

6. Examples

We present three examples to show how our new method performs as well as the SVD approach. The three models of T are adopted from [6]. We wrote a program in MATLAB and ran it on a SUN/Sparc2000 computer. Each estimated eigenvector was computed using seven inverse power iterations [3], and the singular value decomposition was computed using the MATLAB function SVD.

In our examples, we calculate the condition number of T, viz. $\kappa_2(T)$, and determine its numerical rank $n - k$ using the tolerance

$$\tau = 10^{-3}.$$

The matrices are ill-conditioned (calling for regularization), and our new technique always calculates the numerical rank correctly. After computing the rank-revealing decomposition

$$T = VSV^T,$$

where

$$S = \begin{pmatrix} \bar{S} & E \\ E^T & G \end{pmatrix},$$

we partition

$$V = (V_S, \; V_N),$$

where V_N is $n \times k$. Denote by x_S the restored signal vector computed by our new symmetric decomposition. Then

$$x_S = V_S \bar{S}^{-1} V_S^T \hat{x}.$$

Correspondingly, in the SVD

$$T = U \Sigma W^T,$$

we partition

$$W = (W_S, \; W_N) \text{ and } U = (U_S, \; U_N) \,,$$

where W_N and U_N are $n \times k$ matrices; similarly,

$$\Sigma = \mathrm{diag}(\Sigma_S, \; \Sigma_S) \,,$$

where

$$\begin{cases} \Sigma_S = \mathrm{diag}(\sigma_1, \ldots, \sigma_{n-k}) \,, \\ \Sigma_N = \mathrm{diag}(\sigma_{n-k+1}, \ldots, \sigma_n) \,. \end{cases}$$

Since T is symmetric, U and W are the same except for signs in their columns. Let x_T denote the restored signal vector computed by the TSVD method. Then

$$x_T = W_S \Sigma_S^{-1} U_S^T \hat{x}.$$

We use the parameter

$$\epsilon = \|V_N^T W_S\|_2$$

to measure the distance between the two subspaces Range(V_S) and Range(W_S) [3].

Example 1. We pick a 250×250 banded symmetric Toeplitz matrix T:

$$t_m = \begin{cases} 1, & \text{if } m = 1; \\ \dfrac{\sin^2((m-1)/\omega)}{((m-1)/\omega)^2}, & \text{if } 2 \le m \le 5; \\ 0, & \text{otherwise}; \end{cases}$$

with $\omega = 4.0$. The matrix is ill-conditioned:

$$\kappa_2(T) \approx 2.2 \times 10^6 \,.$$

Its numerical rank equals 248, as detected by both methods; hence

$$k = 2\,,$$

a small rank deficiency. After regularization, we get a much better conditioned matrix $\bar{S}$:

$$\kappa_2(\bar{S}) \approx 3.5 \times 10^2.$$

In Table 1, we present the three $(= k+1)$ smallest (in magnitude) eigenvalues λ_i computed by the MATLAB SVD routine and $\hat{\lambda}_i$ computed by our new algorithm.

Since the eigenvalues are well isolated, the estimates are accurate. As predicted by the analysis in Section 5, the off-diagonal blocks E and triu(G) are small:

$$\begin{cases} \|E\|_F = 7.45 \times 10^{-12}\,, \\ \|\mathrm{triu}(G)\|_F = 1.19 \times 10^{-4} \approx \sqrt{\lambda_{249}^2 + \lambda_{250}^2}\,. \end{cases}$$

Furthermore, the error ϵ equals 4.0×10^{-9}, and so the subspace approximation of Range(W_S) by Range(V_S) is also very good.

Example 2. Consider a band Toeplitz matrix T of order 150:

$$t_m = \begin{cases} 2\omega, & \text{if } m = 1; \\ \dfrac{\sin(2\pi\omega(m-1))}{\pi(m-1)}, & \text{if } 2 \le m \le 9; \\ 0, & \text{otherwise}; \end{cases}$$

with $\omega = 0.05$. While the condition number of the given matrix T is large:

$$\kappa_2(T) \approx 1.2 \times 10^6,$$

the condition number of the regularized matrix $\bar{S}$ is acceptable:

$$\kappa_2(\bar{S}) \approx 9.5 \times 10^2.$$

Both methods computed the rank deficiency as

$$k = 5\,.$$

Table 2 lists the six smallest (in magnitude) eigenvalues as computed by the MATLAB routine and as estimated by our procedure.

Since the three smallest eigenvalues are closely clustered, the approximate eigenvalues $\hat{\lambda}_{148}$ and $\hat{\lambda}_{149}$ are inaccurate and

$$\begin{cases} \|E\|_F = 1.36 \times 10^{-1}\,, \\ \|\mathrm{triu}(G)\|_F = 7.46 \times 10^{-3}\,. \end{cases}$$

The error ϵ equals 1.4×10^{-2} and so the estimation of Range(W_S) by Range(V_S) is off.

Example 3. We choose a 120×120 positive definite Toeplitz matrix T:

$$t_m = \begin{cases} \dfrac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(m-1)^2}{2\sigma^2}}, & \text{if } 1 \le m \le 8; \\ 0, & \text{otherwise}; \end{cases}$$

with $\sigma = 2.0$. The matrix T is ill-conditioned:

$$\kappa_2(T) \approx 1.6 \times 10^6.$$

Note that

$$k = 47$$

with both methods and

$$\kappa_2(\bar{S}) \approx 1.1 \times 10^3.$$

Table 1. Three smallest eigenvalues in Example 1.

i	248	249	250
λ	$2.26e-2$	$1.19e-4$	$3.51e-6$
$\hat{\lambda}_i$	$2.17e-2$	$1.19e-4$	$3.51e-6$

Table 2. Six smallest eigenvalues in Example 2.

i	145	146	147	148	149	150
λ_i	$1.19e-3$	$4.88e-4$	$1.71e-4$	$2.76e-5$	$1.00e-5$	$9.37e-6$
$\hat{\lambda}_i$	$1.19e-3$	$4.86e-4$	$1.50e-4$	$5.83e-5$	$5.94e-5$	$9.43e-6$

Figure 2 plots the forty-eight ($= k+1$) smallest eigenvalues computed by the MATLAB SVD routine (represented by a solid line) against those estimated by our symmetric rank-revealing program (represented by a dashed line).

As shown in Figure 2, the eigenvalues decrease gradually and λ_{n-k} is close to λ_{n-k+1}. Just as in Example 2, we observe sizable errors:

$$\begin{cases} \|E\|_F = 5.26 \times 10^{-2}\,, \\ \|\mathrm{triu}(G)\|_F = 3.97 \times 10^{-2}\,. \end{cases}$$

The error ϵ equals 4.8×10^{-1}. So, Range(V_S) does not estimate Range(W_S) very well.

What have we learned from these three examples? The TSVD and our method perform well when there is a significant gap between the large eigenvalues and the small ones (such as in Example 1), and may not work well otherwise. In Example 4, we show that even in the case where the eigenvalues decrease gradually (as in Examples 2 and 3) both the TSVD and our new approach still do good jobs in restoring the original signal vector.

Fig. 3. Original signal vector x (left) and noisy signal vector $\hat{x}$ (right) in Example 4.

Example 4. We stay with the same matrix as in Example 3. In addition, we choose an original signal vector $x \in R^{120}$ given by Nagy [6]. The noise vector w is a random vector generated in MATLAB by a normal distribution with zero mean and unit variance. The noise is scaled so that

$$\|w\|_2 / \|Tx\|_2 = 0.001.$$

We use this ratio as the tolerance τ for the numerical rank. Figure 3 depicts the original signal vector x on the left and the noisy signal vector $\hat{x}$ on the right, and Figure 4 presents the signal vector x_T restored by the TSVD method (left) and vector x_S restored by our method (right). The results show that the restoration capabilities of both methods are comparable.

Fig. 4. Vector x_T restored via TSVD (left) and vector x_S restored via our new scheme (right) in Example 4.

Acknowledgements

The authors acknowledge valuable comments by David Vandevoorde. F.T. Luk was supported in part by the Office of Naval Research under contract N00014-93-1-0268, and S. Qiao by the Natural Sciences and Engineering Research Council of Canada under grant OGP0046301.

References

1. H.C. Andrews and B.R. Hunt, *Digital Image Restoration,* Prentice-Hall, Englewood Cliffs, NJ, 1977.
2. J. Chun, T. Kailath and H. Lev-Ari, "Fast parallel algorithms for QR and triangular factorization," *SIAM J. Sci. Stat. Comput.,* 8, 1987, No. 6, pp. 899-913.
3. G.H. Golub and C.F. Van Loan, *Matrix Computations,* 2nd ed., Johns Hopkins University Press, Baltimore, MD, 1989.
4. P.C. Hansen, "The truncated SVD as a method for regularization," *BIT,* 27, 1987, pp. 534-553.
5. F.T. Luk and D. Vandevoorde, "SVD-based analysis of image boundary distortion," in *SVD and Signal Processing, III: Algorithms, Analysis and Applications*, M. Moonen and B. De Moor, Eds., Elsevier, Amsterdam, 1995, pp. 305-313.
6. J.G. Nagy, "Fast algorithms for the regularization of banded Toeplitz least squares problems," in Proc. SPIE 2296, *Advanced Signal Processing: Algorithms, Architectures, and Implementations V,* F.T. Luk, Ed., SPIE, Bellingham, WA, 1994, pp. 566-575.
7. G.W. Stewart, "An updating algorithm for subspace tracking," *IEEE Trans. Signal Proc.,* 40, 1992, pp. 1535-1541.

Franklin T. Luk is professor and chairman of computer science at Rensselaer Polytechnic Institute in Troy, New York. His reseach interests include scientific computing and signal processing. Dr. Luk received the BS degree (1972) in Mathematics from the California Institute of Technology, the MS degree (1974) in Statistics and the PhD degree (1978) in Computer Science from Stanford University. He taught computer science and electrical engineering at Cornell University from 1978 to 1992, and moved to Rensselaer in 1992 to head the computer science department. He spent sabbatic leaves at the Thinking Machines Corporation (1986-87), and the Chinese University of Hong Kong (1993-94).

Sanzheng Qiao is associate professor of computer science at McMaster University in Hamilton, Ontario, Canada. Dr. Qiao received the BS degree (1966) in Mathematics from Shanghai Teacher's University, China, the MS degree (1986) in Computer Science and the PhD degree (1987) in Applied Mathematics from Cornell University. He has taught at McMaster since 1989, and was on sabbatic leave at the City University of Hong Kong during the 1995-96 academic year.