SVD approximations for large-scale inverse problems

Clarissa Garvey   Chang Meng   James Nagy

Department of Mathematics and Computer Science

EMORY UNIVERSITY

ccgarve@emory.edu

September 12, 2017

Supported by NSF

Singular Value Decomposition (SVD)

A real, $N \times N$ (with $n^2 = N$) matrix K can be decomposed into a singular value decomposition (SVD):

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

where

Singular Value Decomposition (SVD)

A real, $N \times N$ (with $n^2 = N$) matrix K can be decomposed into a singular value decomposition (SVD):

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

where

- $U^T U = UU^T = \mathbb{1}$ (the identity matrix)
- $V^T V = VV^T = \mathbb{1}$
- Σ is a non-negative diagonal matrix with diagonal elements σ_i ordered so that $\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_N$.

Singular Value Decomposition (SVD)

A real, $N \times N$ (with $n^2 = N$) matrix K can be decomposed into a singular value decomposition (SVD):

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

where

- $U^T U = U U^T = \mathbb{1}$ (the identity matrix)
- $V^T V = V V^T = \mathbb{1}$
- Σ is a non-negative diagonal matrix with diagonal elements σ_i ordered so that $\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_N$.

The columns of U and V are the left and right singular vectors of A, respectively.

The σ_i are the singular values of A.

Error Amplification

Our linear problems are of the form

$$\mathbf{b} = K\mathbf{x} + \mathbf{e}$$

where

Error Amplification

Our linear problems are of the form

$$\mathbf{b} = K\mathbf{x} + \mathbf{e}$$

where

- $\mathbf{b} \in \mathbb{R}^N$ is the measured data,
- $\mathbf{x} \in \mathbb{R}^N$ is the desired, unknown solution,
- $K \in \mathbb{R}^{N \times N}$ is a matrix relating the true solution to observation, and
- $\mathbf{e} \in \mathbb{R}^N$ is unknown error.

Error Amplification

Our linear problems are of the form

$$\mathbf{b} = K\mathbf{x} + \mathbf{e}$$

where

- $\mathbf{b} \in \mathbb{R}^N$ is the measured data,
- $\mathbf{x} \in \mathbb{R}^N$ is the desired, unknown solution,
- $K \in \mathbb{R}^{N \times N}$ is a matrix relating the true solution to observation, and
- $\mathbf{e} \in \mathbb{R}^N$ is unknown error.

If the singular values of K tend to zero without a gap, K is ill-conditioned.

Error Amplification

Our linear problems are of the form

$$\mathbf{b} = K\mathbf{x} + \mathbf{e}$$

where

- $\mathbf{b} \in \mathbb{R}^N$ is the measured data,
- $\mathbf{x} \in \mathbb{R}^N$ is the desired, unknown solution,
- $K \in \mathbb{R}^{N \times N}$ is a matrix relating the true solution to observation, and
- $\mathbf{e} \in \mathbb{R}^N$ is unknown error.

If the singular values of K tend to zero without a gap, K is ill-conditioned.

Then direct solution using the SVD may amplify noise:

$$K^{-1}\mathbf{b} = \mathbf{x} + K^{-1}\mathbf{e} = \mathbf{x} + V\Sigma^{-1}U^T\mathbf{e}$$

Filtering

Filtering can avoid this issue.

Filtering

Filtering can avoid this issue.

$$\begin{aligned} K^{-1} &= V\Sigma^{-1}U^T \\ &= Vdiag\left(\frac{1}{\sigma_1}, ..., \frac{1}{\sigma_N}\right)U^T \end{aligned}$$

Filtering

Filtering can avoid this issue.

$$
\begin{aligned}
K^{-1} &= V\Sigma^{-1}U^T \\
&= V diag\left(\frac{1}{\sigma_1}, \ldots, \frac{1}{\sigma_N}\right) U^T
\end{aligned}
$$

$$
K_F^{-1} = V diag\left(\frac{\phi_1}{\sigma_1}, \ldots, \frac{\phi_N}{\sigma_N}\right) U^T
$$

Filtering

Filtering can avoid this issue.

$$K^{-1} = V\Sigma^{-1}U^T$$
$$= Vdiag\left(\frac{1}{\sigma_1}, ..., \frac{1}{\sigma_N}\right)U^T$$

$$K_F^{-1} = Vdiag\left(\frac{\phi_1}{\sigma_1}, ..., \frac{\phi_N}{\sigma_N}\right)U^T$$

$\{\phi_i\}$ are weights such that, for small σ_i, $\phi_i = 0$.

In a truncated SVD, $\phi_i = 0$ if $i > k$, a cutoff index.

The Kronecker Product

The Kronecker product of matrices $A_{m \times m}$ and B is

$$A \otimes B = \begin{bmatrix} a_{1,1}B & \dots & a_{1,m}B \\ \vdots & & \vdots \\ a_{m,1}B & \dots & a_{m,m}B \end{bmatrix}$$

The Kronecker Product

The Kronecker product of matrices $A_{m\times m}$ and B is

$$A \otimes B = \begin{bmatrix} a_{1,1}B & \dots & a_{1,m}B \\ \vdots & & \vdots \\ a_{m,1}B & \dots & a_{m,m}B \end{bmatrix}$$

Given $A = U_A \Sigma_A V_A^T$ and $B = U_B \Sigma_B V_B^T$,

The Kronecker Product

The Kronecker product of matrices $A_{m\times m}$ and B is

$$A \otimes B = \begin{bmatrix} a_{1,1}B & \dots & a_{1,m}B \\ \vdots & & \vdots \\ a_{m,1}B & \dots & a_{m,m}B \end{bmatrix}$$

Given $A = U_A \Sigma_A V_A^T$ and $B = U_B \Sigma_B V_B^T$,

$$A \otimes B = (U_A \otimes U_B)(\Sigma_A \otimes \Sigma_B)(V_A \otimes V_B)^T$$

Decomposition via Kronecker Product Summation

Arbitrary $N \times N$ matrices may not be writeable as $A_{n\times n} \otimes B_{n\times n}$

Decomposition via Kronecker Product Summation

Arbitrary $N \times N$ matrices may not be writeable as $A_{n\times n} \otimes B_{n\times n}$

But all $N \times N$ matrices K can be represented as

$$K = \sum_{i=1}^{R} A_i \otimes B_i$$

where all A_i, B_i are $n \times n$ with $n^2 = N$ in size and the index R is the Kronecker rank. (See *Van Loan and Pitsiannis, 1993 [**?**]*)

SVD Approx. via Kronecker Products

Suppose we have $K = \sum_{i=1}^{R} A_i \otimes B_i$, and

SVD Approx. via Kronecker Products

Suppose we have $K = \sum_{i=1}^{R} A_i \otimes B_i$, and

- $A_1 = U_{A1} \Sigma_{A1} V_{A1}^T$
- $B_1 = U_{B1} \Sigma_{B1} V_{B1}^T$

SVD Approx. via Kronecker Products

Suppose we have $K = \sum_{i=1}^{R} A_i \otimes B_i$, and

- $A_1 = U_{A1} \Sigma_{A1} V_{A1}^T$
- $B_1 = U_{B1} \Sigma_{B1} V_{B1}^T$
- $U_1 = U_{A1} \otimes U_{B1}$
- $\Sigma_1 = \Sigma_{A1} \otimes \Sigma_{B1}$
- $V_1 = V_{A1} \otimes V_{B1}$

SVD Approx. via Kronecker Products

Suppose we have $K = \sum_{i=1}^{R} A_i \otimes B_i$, and

- $A_1 = U_{A1} \Sigma_{A1} V_{A1}^T$
- $B_1 = U_{B1} \Sigma_{B1} V_{B1}^T$
- $U_1 = U_{A1} \otimes U_{B1}$
- $\Sigma_1 = \Sigma_{A1} \otimes \Sigma_{B1}$
- $V_1 = V_{A1} \otimes V_{B1}$

Then $A_1 \otimes B_1 = U_1 \Sigma_1 V_1^T$.

SVD Approx. via Kronecker Products

Suppose we have $K = \sum_{i=1}^{R} A_i \otimes B_i$, and

- $A_1 = U_{A1} \Sigma_{A1} V_{A1}^T$
- $B_1 = U_{B1} \Sigma_{B1} V_{B1}^T$
- $U_1 = U_{A1} \otimes U_{B1}$
- $\Sigma_1 = \Sigma_{A1} \otimes \Sigma_{B1}$
- $V_1 = V_{A1} \otimes V_{B1}$

Then $A_1 \otimes B_1 = U_1 \Sigma_1 V_1^T$.

Further, $K = U_1 \Sigma_1 V_1^T + \sum_{i=2}^{R} U_1 U_1^T K_i V_1 V_1^T$

SVD Approx. via Kronecker Products

Suppose we have $K = \sum_{i=1}^{R} A_i \otimes B_i$, and

- $A_1 = U_{A1} \Sigma_{A1} V_{A1}^T$
- $B_1 = U_{B1} \Sigma_{B1} V_{B1}^T$
- $U_1 = U_{A1} \otimes U_{B1}$
- $\Sigma_1 = \Sigma_{A1} \otimes \Sigma_{B1}$
- $V_1 = V_{A1} \otimes V_{B1}$

Then $A_1 \otimes B_1 = U_1 \Sigma_1 V_1^T$.

$$
\begin{aligned}
\text{Further, } K &= U_1 \Sigma_1 V_1^T + \sum_{i=2}^{R} U_1 U_1^T K_i V_1 V_1^T \\
&= U_1 \left(\Sigma_1 + \sum_{i=2}^{R} U_1^T K_i V_1 \right) V_1^T
\end{aligned}
$$

See *Nagy, Ng, and Perrone (2006) [**?**]* and *Kamm and Nagy (2000) [**?**]*

Algorithm Derivation

$$K = U_1 \left(\Sigma_1 + \sum_{i=2}^{R} U_1^T K_i V_1 \right) V_1^T$$

Algorithm Derivation

$$K = U_1 \left(\Sigma_1 + \sum_{i=2}^{R} U_1^T K_i V_1 \right) V_1^T$$

$$= U_1 \left(\begin{bmatrix} \Sigma_{1,k} & 0 \\ 0 & \widehat{\Sigma}_0 \end{bmatrix} + \begin{bmatrix} W_{11} & W_{12} \\ W_{21} & W_{22} \end{bmatrix} \right) V_1^T$$

Algorithm Derivation

$$K = U_1 \left(\Sigma_1 + \sum_{i=2}^{R} U_1^T K_i V_1 \right) V_1^T$$

$$= U_1 \left(\begin{bmatrix} \Sigma_{1,k} & 0 \\ 0 & \widehat{\Sigma}_0 \end{bmatrix} + \begin{bmatrix} W_{11} & W_{12} \\ W_{21} & W_{22} \end{bmatrix} \right) V_1^T$$

$$= U_1 \left(\begin{bmatrix} \Sigma_{1,k} + W_{11} & 0 \\ 0 & \widehat{\Sigma}_0 \end{bmatrix} + \begin{bmatrix} 0 & W_{12} \\ W_{21} & W_{22} \end{bmatrix} \right) V_1^T$$

Algorithm Derivation

$$K = U_1 \left(\Sigma_1 + \sum_{i=2}^{R} U_1^T K_i V_1 \right) V_1^T$$

$$= U_1 \left(\begin{bmatrix} \Sigma_{1,k} & 0 \\ 0 & \widehat{\Sigma}_0 \end{bmatrix} + \begin{bmatrix} W_{11} & W_{12} \\ W_{21} & W_{22} \end{bmatrix} \right) V_1^T$$

$$= U_1 \left(\begin{bmatrix} \Sigma_{1,k} + W_{11} & 0 \\ 0 & \widehat{\Sigma}_0 \end{bmatrix} + \begin{bmatrix} 0 & W_{12} \\ W_{21} & W_{22} \end{bmatrix} \right) V_1^T$$

$$= U_1 \left(\begin{bmatrix} U_k \widehat{\Sigma}_k V_k^T & 0 \\ 0 & \widehat{\Sigma}_0 \end{bmatrix} + \begin{bmatrix} 0 & W_{12} \\ W_{21} & W_{22} \end{bmatrix} \right) V_1^T$$

Algorithm Derivation

$$K = U_1 \left(\begin{bmatrix} U_k \widehat{\Sigma}_k V_k^T & 0 \\ 0 & \widehat{\Sigma}_0 \end{bmatrix} + \begin{bmatrix} 0 & W_{12} \\ W_{21} & W_{22} \end{bmatrix} \right) V_1^T$$

Algorithm Derivation

$$K = U_1 \left(\begin{bmatrix} U_k \widehat{\Sigma}_k V_k^T & 0 \\ 0 & \widehat{\Sigma}_0 \end{bmatrix} + \begin{bmatrix} 0 & W_{12} \\ W_{21} & W_{22} \end{bmatrix} \right) V_1^T$$

$$= U_1 \begin{bmatrix} U_k & 0 \\ 0 & \mathbb{1} \end{bmatrix} \left(\begin{bmatrix} \widehat{\Sigma}_k & 0 \\ 0 & \widehat{\Sigma}_0 \end{bmatrix} + \begin{bmatrix} 0 & U_k^T W_{12} \\ W_{21} V_k & W_{22} \end{bmatrix} \right) \begin{bmatrix} V_k^T & 0 \\ 0 & \mathbb{1} \end{bmatrix} V_1^T$$

Algorithm Derivation

$$K = U_1 \left(\begin{bmatrix} U_k \widehat{\Sigma}_k V_k^T & 0 \\ 0 & \widehat{\Sigma}_0 \end{bmatrix} + \begin{bmatrix} 0 & W_{12} \\ W_{21} & W_{22} \end{bmatrix} \right) V_1^T$$

$$= U_1 \begin{bmatrix} U_k & 0 \\ 0 & \mathbb{1} \end{bmatrix} \left(\begin{bmatrix} \widehat{\Sigma}_k & 0 \\ 0 & \widehat{\Sigma}_0 \end{bmatrix} + \begin{bmatrix} 0 & U_k^T W_{12} \\ W_{21} V_k & W_{22} \end{bmatrix} \right) \begin{bmatrix} V_k^T & 0 \\ 0 & \mathbb{1} \end{bmatrix} V_1^T$$

$$= \widehat{U} \left(\begin{bmatrix} \widehat{\Sigma}_k & 0 \\ 0 & \widehat{\Sigma}_0 \end{bmatrix} + \begin{bmatrix} 0 & \widehat{W}_{12} \\ \widehat{W}_{21} & W_{22} \end{bmatrix} \right) \widehat{V}^T.$$

Algorithm

Given

$$K = \widehat{U}\left(\begin{bmatrix}\widehat{\Sigma}_k & 0 \\ 0 & \widehat{\Sigma}_0\end{bmatrix} + \begin{bmatrix}0 & \widehat{W_{12}} \\ \widehat{W_{21}} & W_{22}\end{bmatrix}\right)\widehat{V}^T$$

Algorithm

Given

$$K = \widehat{U} \left(\begin{bmatrix} \widehat{\Sigma}_k & 0 \\ 0 & \widehat{\Sigma}_0 \end{bmatrix} + \begin{bmatrix} 0 & \widehat{W_{12}} \\ \widehat{W_{21}} & W_{22} \end{bmatrix} \right) \widehat{V}^T$$

partition $\widehat{U} = \begin{bmatrix} \widehat{U}_k & \widehat{U}_0 \end{bmatrix}$ and $V = \begin{bmatrix} \widehat{V}_k & \widehat{V}_0 \end{bmatrix}$.

Algorithm

Given

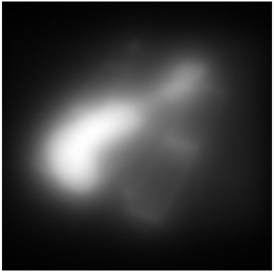
$$K = \widehat{U} \left(\begin{bmatrix} \widehat{\Sigma}_k & 0 \\ 0 & \widehat{\Sigma}_0 \end{bmatrix} + \begin{bmatrix} 0 & \widehat{W}_{12} \\ \widehat{W}_{21} & W_{22} \end{bmatrix} \right) \widehat{V}^T$$

partition $\widehat{U} = \begin{bmatrix} \widehat{U}_k & \widehat{U}_0 \end{bmatrix}$ and $V = \begin{bmatrix} \widehat{V}_k & \widehat{V}_0 \end{bmatrix}$.

Then we can construct a truncated SVD

$$K \approx \widehat{U}_k \widehat{\Sigma}_k \widehat{V}_k^T$$

Image Deblurring Example



observed image

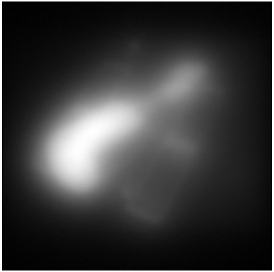
SVD approx
1 Kron term

SVD approx
5 Kron terms

In this example:

- K is $N \times N$, with $N = 65,536 \times 65,536$.
- K_{TSVD} uses $k = 2,601$ for truncation.

Image Deblurring Example



observed image

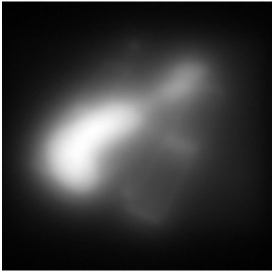
SVD approx
1 Kron term

SVD approx
10 Kron terms

In this example:

- K is $N \times N$, with $N = 65,536 \times 65,536$.
- K_{TSVD} uses $k = 2,601$ for truncation.

Image Deblurring Example



observed image

SVD approx

1 Kron term

SVD approx

67 Kron terms

In this example:

- K is $N \times N$, with $N = 65,536 \times 65,536$.
- K_{TSVD} uses $k = 2,601$ for truncation.

Conclusion

- We created an algorithm for approximating a truncated SVD.
- Kronecker products help keep computational costs down.
- We can compute the TSVD on larger problems than traditional methods allow.
- This method is fast (seconds) and produces reasonable solutions.

References and Acknowledgement

- C.F. Van Loan and N. Pitsianis. Approximation with Kronecker Products. *Linear Algebra for Large Scale and Real-time Applications, Springer Netherlands* 293-314, 1993.
- J. Nagy, M.K. Ng, L. Perrone. Kronecker Product Approximations for Image Restoration with Reflexive Boundary Conditions. *SIAM Journal on Matrix Analysis and Applications*, 25(3), 829-841, 2006.
- J. Kamm, and J. Nagy. Optimal Kronecker Product Approximation of Block Toeplitz Matrices. *SIAM Journal on Matrix Analysis and Applications* 22.1, 155-172, 2000.
- N. Halko, P.G. Martinsson, and J.A. Tropp. Finding Structure with Randomness: Probabilistic Algorithms for Constructing Approximate Matrix Decompositions. *SIAM Review* 53.2, 217-288, 2011.

Acknowledgement: This work is supported by the National Science Foundation under grant number DMS-1522760.

Thank you!