SVD Based Image Compression using Lanczos and QR Algorithms

Aarush Dilawri
Department of Electrical Engineering
Indian Institute of Technology
Course: EE1030 Matrix Theory

1. Summary of Strang's Video

Professor Gilbert Strang's lecture on the Singular Value Decomposition (SVD) explains that every matrix can be factored into three simple transformations:

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

where $\mathbf{U}$ and $\mathbf{V}$ are orthogonal, and $\mathbf{\Sigma}$ is diagonal with non-negative singular values $\sigma_1 \geq \sigma_2 \geq \cdots \geq 0$. This is similar to Eigenvalue decomposition but in that, the matrix $\mathbf{A}$ needss to be positive, definite and symmetric. But SVD works for any matrix. Geometrically, $\mathbf{V}$ represents a rotation in the input space, $\mathbf{\Sigma}$ scales along orthogonal directions, and $\mathbf{U}$ represents another rotation to the output space.

Strang emphasizes that small singular values correspond to directions of little importance. Ignoring them gives a low-rank approximation that retains the main structure of the data—this idea forms the basis for image compression.

2. Image Compression using SVD

Let an image be represented by a matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$, where each entry denotes the grayscale intensity. By SVD,

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T = \sum_{i=1}^{r} \sigma_i \mathbf{u}_i \mathbf{v}_i^T,$$

where $r = \text{rank}(\mathbf{A})$. Each term $\sigma_i \mathbf{u}_i \mathbf{v}_i^T$ is a **rank-1 matrix**. If we keep only the largest k singular values,

$$\mathbf{A}_k = \sum_{i=1}^{k} \sigma_i \mathbf{u}_i \mathbf{v}_i^T,$$

then $\mathbf{A}_k$ is the best rank-k approximation to $\mathbf{A}$ in Frobenius norm:

$$\mathbf{A}_k = \underset{\text{rank}(\mathbf{B})=k}{\arg\min} \|\mathbf{A} - \mathbf{B}\|_F.$$

This means that if we want to express our matrix $\mathbf{A}$ as a rank k matrix, then the best approximation will be provided by SVD.The Frobenius norm will be minimum for the matrix formed by SVD. Thus, we can store $\mathbf{U}_k, \mathbf{\Sigma}_k, \mathbf{V}_k$ (each of size k) instead of $\mathbf{A}$, reducing storage from mn to $k(m+n+1)$ numbers.

3. Derivation of SVD from Eigenvalue Decomposition

We start with

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T.$$

Then

$$\mathbf{A}^T\mathbf{A} = \mathbf{V}\mathbf{\Sigma}^T\mathbf{\Sigma}\mathbf{V}^T = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T,$$

where $\mathbf{\Lambda} = \text{diag}(\sigma_1^2, \ldots, \sigma_r^2)$. On comparing this with Eigenvalue Decomposition of $\mathbf{A}^T\mathbf{A}$, we get columns of $\mathbf{V}$ as the eigenvectors of $\mathbf{A}^T\mathbf{A}$ and

$$\mathbf{A}^T\mathbf{A}\mathbf{v}_i = \sigma_i^2\mathbf{v}_i.$$

Once $\mathbf{v}_i$ and σ_i are known,

$$\mathbf{u}_i = \frac{\mathbf{A}\mathbf{v}_i}{\sigma_i}.$$

Hence, computing SVD reduces to diagonalizing $\mathbf{A}^T\mathbf{A}$.

4. Krylov Subspace and the Lanczos Reduction

Let $\mathbf{G} = \mathbf{A}^T\mathbf{A}$ (symmetric and positive semi-definite). We define the level k **Krylov subspace** generated by $\mathbf{G}$ and a vector $\mathbf{q}_1$:

$$\mathcal{K}_k(\mathbf{G}, \mathbf{q}_1) = \text{span}\{\mathbf{q}_1, \mathbf{G}\mathbf{q}_1, \mathbf{G}^2\mathbf{q}_1, \ldots, \mathbf{G}^{k-1}\mathbf{q}_1\}.$$

The Lanczos algorithm constructs an orthonormal basis $\mathbf{Q}_k = [\mathbf{q}_1, \ldots, \mathbf{q}_k]$ of this subspace such that

$$\mathbf{Q}_k^T\mathbf{G}\mathbf{Q}_k = \mathbf{T}_k,$$

where $\mathbf{T}_k$ is symmetric and tridiagonal.

Lanczos Recurrence

Starting with a random unit vector $\mathbf{q}_1$,

$$\begin{aligned} \alpha_i &= \mathbf{q}_i^T\mathbf{G}\mathbf{q}_i, \\ \mathbf{r}_i &= \mathbf{G}\mathbf{q}_i - \beta_{i-1}\mathbf{q}_{i-1}, -\alpha_i\mathbf{q}_i, \\ \beta_i &= \|\mathbf{r}_i\|_2, \\ \mathbf{q}_{i+1} &= \frac{\mathbf{r}_i}{\beta_i}. \end{aligned} \tag{1}$$

Hence:

$$\mathbf{G}\mathbf{q}_i = \beta_{i-1}\mathbf{q}_{i-1} + \alpha_i\mathbf{q}_i + \beta_i\mathbf{q}_{i+1}.$$

Stacking these for $i = 1, \ldots, k$ gives:

$$\mathbf{G}\mathbf{Q}_k = \mathbf{Q}_k\mathbf{T}_k + \beta_k\mathbf{q}_{k+1}\mathbf{e}_k^T.$$

Thus:

$$\mathbf{T}_k = \mathbf{Q}_k^T\mathbf{G}\mathbf{Q}_k.$$

$\mathbf{T}_k$ is tridiagonal with diagonals α_i and sub/super-diagonals β_i.

5. Rayleigh Quotient and Ritz Approximations

For a symmetric $\mathbf{G}$, the Rayleigh quotient is

$$\rho(\mathbf{x}) = \frac{\mathbf{x}^T \mathbf{G} \mathbf{x}}{\mathbf{x}^T \mathbf{x}}.$$

If $\mathbf{x}$ is an eigenvector, then $\rho(\mathbf{x})$ equals its eigenvalue.

Let $\mathbf{x} = \mathbf{Q}_k \mathbf{y}$ where $\mathbf{Q}_k$ spans $\mathcal{K}_k(\mathbf{G}, \mathbf{q}_1)$. Then:

$$\rho(\mathbf{Q}_k \mathbf{y}) = \frac{\mathbf{y}^T (\mathbf{Q}_k^T \mathbf{G} \mathbf{Q}_k) \mathbf{y}}{\mathbf{y}^T \mathbf{y}} = \frac{\mathbf{y}^T \mathbf{T}_k \mathbf{y}}{\mathbf{y}^T \mathbf{y}}.$$

Therefore, eigenvalues of $\mathbf{T}_k$ (called **Ritz values**) minimize or maximize the Rayleigh quotient within the Krylov subspace. They approximate the true eigenvalues of $\mathbf{G}$, with increasing accuracy as k grows.

6. Householder Reflectors and QR Factorization

To reduce a matrix $\mathbf{A}$ to upper-triangular form, we construct orthogonal reflectors. Given $\mathbf{x} \in \mathbb{R}^m$:

$$\mathbf{v} = \mathbf{x} + \text{sign}(x_1) \|\mathbf{x}\| \mathbf{e}_1, \tag{2}$$

$$\mathbf{u} = \frac{\mathbf{v}}{\|\mathbf{v}\|}, \tag{3}$$

$$\mathbf{H} = \mathbf{I} - 2\mathbf{u}\mathbf{u}^T. \tag{4}$$

Then:

$$\mathbf{H}\mathbf{x} = \mathbf{x} - 2\mathbf{u}(\mathbf{u}^T \mathbf{x}) = -\text{sign}(x_1) \|\mathbf{x}\| \mathbf{e}_1,$$

and

$$\mathbf{H}^T \mathbf{H} = \mathbf{I}.$$

Successive $\mathbf{H}_i$ eliminate elements below the diagonal, giving:

$$\mathbf{A} = \mathbf{Q}\mathbf{R}, \qquad \mathbf{Q} = \mathbf{H}_1 \mathbf{H}_2 \dots \mathbf{H}_p.$$

7. QR Iteration and Matrix Similarity

For $\mathbf{T}$ symmetric tridiagonal, perform:

$$\mathbf{T} = \mathbf{Q}\mathbf{R}, \quad \mathbf{T}' = \mathbf{R}\mathbf{Q}.$$

Since $\mathbf{Q}$ is orthogonal:

$$\mathbf{T}' = \mathbf{Q}^T \mathbf{T} \mathbf{Q}.$$

Hence $\mathbf{T}$ and $\mathbf{T}'$ are **similar** and share eigenvalues. Repeatedly applying QR reduces $\mathbf{T}$ to diagonal form:

$$\lim_{i \to \infty} \mathbf{T}^{(i)} = \text{diag}(\lambda_1, \dots, \lambda_k),$$

where λ_i are eigenvalues of $\mathbf{G}$.

8. Wilkinson Shift

To accelerate convergence, a scalar shift μ is introduced:

$$\mathbf{T} - \mu\mathbf{I} = \mathbf{QR}, \qquad \mathbf{T}_{\text{new}} = \mathbf{RQ} + \mu\mathbf{I}.$$

For the trailing 2×2 block

$$\begin{bmatrix} a & b \\ b & d \end{bmatrix},$$

the shift is

$$\mu = d - \frac{b^2}{|a-d| + \sqrt{(a-d)^2 + b^2}}.$$

This choice ensures that the smallest off-diagonal term converges rapidly to zero.

9. Relation Between T and $\mathbf{A}^T\mathbf{A}$

From the Lanczos recurrence:

$$\mathbf{GQ}_k = \mathbf{Q}_k\mathbf{T}_k + \beta_k\mathbf{q}_{k+1}\mathbf{e}_k^T.$$

Premultiplying by $\mathbf{Q}_k^T$:

$$\mathbf{Q}_k^T\mathbf{GQ}_k = \mathbf{T}_k.$$

Thus, $\mathbf{T}_k$ and $\mathbf{G}$ are similar in the subspace $\mathcal{K}_k$, meaning their eigenvalues approximate each other. If $\mathbf{T}_k\mathbf{y}_i = \lambda_i\mathbf{y}_i$, then

$$\mathbf{G}(\mathbf{Q}_k\mathbf{y}_i) \approx \lambda_i(\mathbf{Q}_k\mathbf{y}_i),$$

so approximate eigenvectors of $\mathbf{G}$ are given by $\mathbf{v}_i = \mathbf{Q}_k\mathbf{y}_i$.

The singular values are then $\sigma_i = \sqrt{\lambda_i}$, and

$$\mathbf{u}_i = \frac{\mathbf{A}\mathbf{v}_i}{\sigma_i}.$$

10. Complete Algorithm

Input: $\mathbf{A} \in \mathbb{R}^{m\times n}$, target rank k.

1. $\mathbf{G} = \mathbf{A}^T\mathbf{A}$.
2. Initialize $\mathbf{q}_1 = \frac{1}{\sqrt{n}}\mathbf{1}$.
3. Apply Lanczos iterations to get $\mathbf{Q}_k, \mathbf{T}_k$.
4. Diagonalize $\mathbf{T}_k = \mathbf{Y}\mathbf{\Lambda}\mathbf{Y}^T$ via shifted QR.
5. $\mathbf{V}_k = \mathbf{Q}_k\mathbf{Y}, \quad \mathbf{\Sigma}_k = \sqrt{\mathbf{\Lambda}}$.
6. $\mathbf{U}_k = \mathbf{A}\mathbf{V}_k\mathbf{\Sigma}_k^{-1}$.
7. Reconstruct $\mathbf{A}_k = \mathbf{U}_k\mathbf{\Sigma}_k\mathbf{V}_k^T$.

11. Error Analysis

Frobenius error:

$$E_k = \|\mathbf{A} - \mathbf{A}_k\|_F, \quad \epsilon_k = \frac{E_k}{\|\mathbf{A}\|_F}.$$

Since $\mathbf{A}_k$ minimizes $\|\mathbf{A} - \mathbf{B}\|_F$ over rank-k matrices,

$$E_k^2 = \sum_{i=k+1}^{r} \sigma_i^2.$$

Hence, increasing k strictly decreases the reconstruction error.

12. Pseudocode of the Implemented Algorithm

Algorithm 1: Approximate SVD via Lanczos and QR Iteration

Input: $A \in \mathbb{R}^{m \times n}$, target rank k
Output: Approximate SVD components U_k, Σ_k, V_k

1. Initialize:
Compute $G \leftarrow A^\top A$
Pick a random unit vector q_1

2. Build Krylov subspace (Lanczos iteration):
for $i = 1$ **to** k **do**
  $\alpha_i \leftarrow q_i^\top G q_i$
  $r \leftarrow G q_i - \beta_{i-1} q_{i-1} - \alpha_i q_i$
  $\beta_i \leftarrow \|r\|$
  $q_{i+1} \leftarrow r/\beta_i$
Form tridiagonal matrix T using α_i and β_i

3. Diagonalize T (QR iteration with shift):
repeat
  $\mu \leftarrow$ Wilkinson shift of T
  $[Q, R] \leftarrow \mathrm{QR}(T - \mu I)$
  $T \leftarrow RQ + \mu I$
until *off-diagonal elements* ≈ 0;

4. Compute eigenvalues and singular values:
$\lambda_i \leftarrow \mathrm{diag}(T)$
$\sigma_i \leftarrow \sqrt{\lambda_i}$

5. Compute singular vectors:
$V \leftarrow Q\times$ (eigenvectors of T)
$U \leftarrow (AV)/\Sigma$

6. Reconstruct approximate matrix:
$A_k \leftarrow U_k \Sigma_k V_k^\top$

13. Algorithm Comparison

The table below compares the main algorithms used to compute eigenvalues and singular values. The implemented method, **Lanczos + QR + Wilkinson Shift**, provides the

best balance of speed, numerical stability, and memory efficiency.

Algorithm	**Time Complexity**	**Convergence Rate**	**Stability**	**Memory**	**Remarks**
Full SVD (Golub–Reinsch)	$O(n^3)$	Quadratic	Very High	$O(n^2)$	Classical dense SVD used in LAPACK. Accurate but too slow for large matrices or images.
Jacobi Diagonalization	$O(n^3)$	Linear–Quadratic	Excellent	$O(n^2)$	Eliminates off-diagonal entries using plane rotations. Accurate but extremely slow.
Power Iteration	$O(kn^2)$	Linear	Moderate	$O(n)$	Computes only the dominant eigenpair. Needs deflation for others, making it inefficient.
QR Algorithm (Unshifted)	$O(n^3)$	Slow	High	$O(n^2)$	Repeated QR decompositions; converges slowly for close eigenvalues.
Lanczos (Unreorthogonalized)	$O(kn^2)$	Fast Initially	Poor	$O(kn)$	Quickly builds tridiagonal $\mathbf{T}$ but loses orthogonality, producing "ghost" eigenvalues.
Lanczos + QR (No Shift)	$O(kn^2)$	Moderate	Good	$O(kn)$	Combines tridiagonalization and QR iteration. Stable, but slower than shifted variants.
Lanczos + QR + Wilkinson Shift (Implemented)	$\boldsymbol{O(kn^2)}$	**Fastest (Cubic near convergence)**	**High**	$\boldsymbol{O(kn)}$	**Chosen method.** Uses tridiagonal reduction and shifted QR for rapid, stable convergence in image compression.

Table 1: Comparison of different algorithms for computing singular values and eigenvalues.

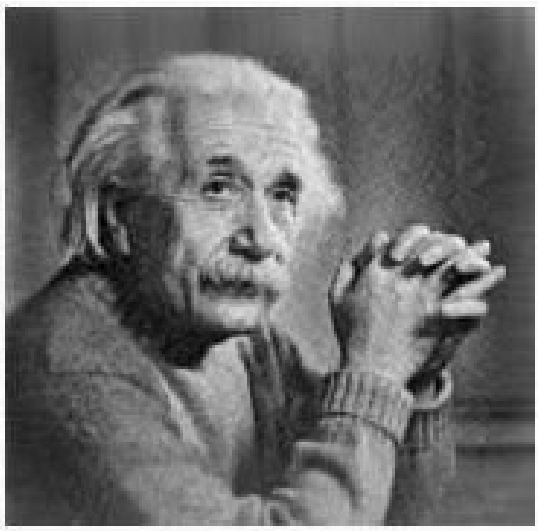
14. Reconstructed Images for Different k

Below are the results for different images and k values (the paths will be inserted later):

(a) Globe, $k = 50$

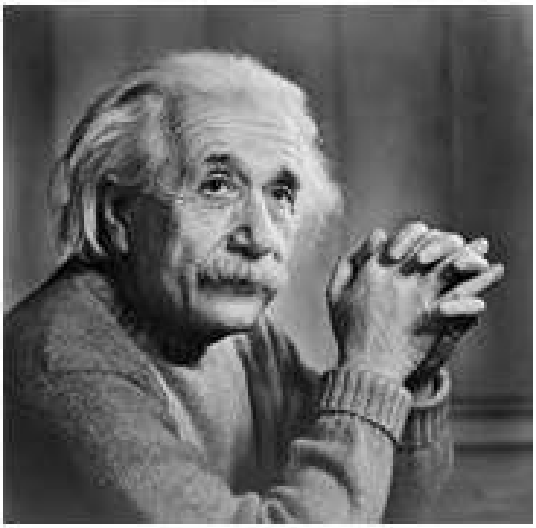
(b) Globe, $k = 100$

(c) Globe, $k = 800$

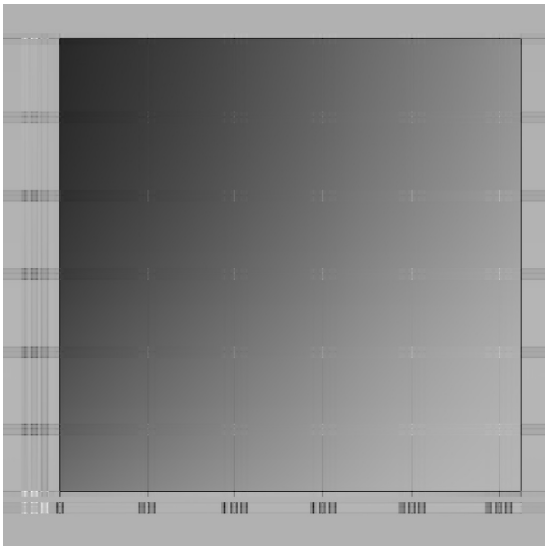


(d) Einstein, $k = 50$

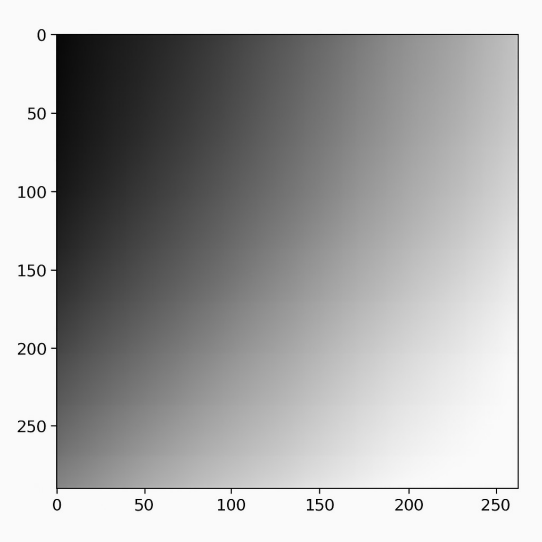
Figure 1: Reconstructed images using truncated SVD for different k.



(a) Einstein, $k = 170$



(b) Greyscale, $k = 5$



(c) Greyscale, $k = 100$

(d) Avenger, $k = 100$

Figure 2: Reconstructed images using truncated SVD for different k.

---

15. Error Analysis (Experimental Results)

$$E_k = \|\mathbf{A} - \mathbf{A}_k\|_F, \quad \epsilon_k = \frac{E_k}{\|\mathbf{A}\|_F}.$$

k	E_k (Frobenius Norm)	Relative Error ϵ_k
50	7.19×10^3	0.045
100	4.33×10^3	0.027
400	7.61×10^2	0.0048
800	1.902	$1.2017 \times 10^{-}5$

---

16. Trade-offs and Reflections

- **Accuracy vs. Speed:** Increasing k improves accuracy but increases computation time.
- **Storage Efficiency:** Low-rank SVD drastically reduces storage for large matrices.
- **Orthogonality:** Without reorthogonalization, $\mathbf{Q}_k^T \mathbf{Q}_k \neq \mathbf{I}$, degrading accuracy.
- **Shifted QR:** Wilkinson shift improves convergence speed significantly.
- **Implementation Stability:** Setting iteration limits prevents divergence.
- **Insight:** Coding Lanczos and QR from scratch provided deep intuition into how eigen decompositions actually converge numerically.

—

17. Observations

- Loss of orthogonality in $\mathbf{Q}_k$ leads to inaccurate Ritz values; hence reorthogonalization is necessary.
- Limiting iteration count prevents divergence in QR.
- Lanczos reduction drastically lowers the dimensionality of the eigenproblem.

18. Conclusion

The Lanczos + QR method with Wilkinson shift provides a fast, memory-efficient, and stable approach to compute SVD for large images. It yields visually accurate reconstructions at a fraction of the computational cost of full SVD, while maintaining strong theoretical foundations in matrix similarity, Krylov subspaces, and orthogonal projections.

19. References

1. Matrix Computations 4th edition by Gene H. Golub and Charles F. Van Loan
2. Linear Algebra and it's applications by Gilbert Strang