Applied Linear Algebra in Image Compression

By: Angela Cao
Mentor: Dhwanit Agarwal
Directed Reading Program (DRP)

Objective Goals

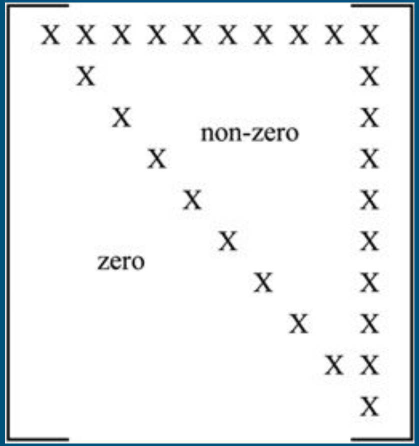
- An overview of the Applied Linear Algebra concepts used in the research project
    - QR Factorization
    - Singular Value Decomposition (SVD)
- An overview of Image Compression as a real-life application of numerical linear algebra

QR Factorization

Suppose A is the original matrix, then A = QR.

[A] = [Q]

Orthogonal Matrix
Done by Gram-
Schmidt



Upper Triangular Matrix

Gram-Schmidt

- Algorithm used to orthogonalize a set of vectors, especially in a matrix
    - Orthogonal Matrix = for matrix Q, $Q^TQ = QQ^T = I$
    - You can prove this!
- In QR Factorization, this concerns the Q matrix.
- Formula

$$
\begin{aligned}
\mathbf{u}_1 &= \mathbf{v}_1, & \mathbf{e}_1 &= \frac{\mathbf{u}_1}{\|\mathbf{u}_1\|} \\
\mathbf{u}_2 &= \mathbf{v}_2 - \mathrm{proj}_{\mathbf{u}_1}(\mathbf{v}_2), & \mathbf{e}_2 &= \frac{\mathbf{u}_2}{\|\mathbf{u}_2\|} \\
\mathbf{u}_3 &= \mathbf{v}_3 - \mathrm{proj}_{\mathbf{u}_1}(\mathbf{v}_3) - \mathrm{proj}_{\mathbf{u}_2}(\mathbf{v}_3), & \mathbf{e}_3 &= \frac{\mathbf{u}_3}{\|\mathbf{u}_3\|} \\
\mathbf{u}_4 &= \mathbf{v}_4 - \mathrm{proj}_{\mathbf{u}_1}(\mathbf{v}_4) - \mathrm{proj}_{\mathbf{u}_2}(\mathbf{v}_4) - \mathrm{proj}_{\mathbf{u}_3}(\mathbf{v}_4), & \mathbf{e}_4 &= \frac{\mathbf{u}_4}{\|\mathbf{u}_4\|} \\
&\vdots & &\vdots \\
\mathbf{u}_k &= \mathbf{v}_k - \sum_{j=1}^{k-1} \mathrm{proj}_{\mathbf{u}_j}(\mathbf{v}_k), & \mathbf{e}_k &= \frac{\mathbf{u}_k}{\|\mathbf{u}_k\|}.
\end{aligned}
$$

$$\mathrm{proj}_{\mathbf{u}}(\mathbf{v}) = \frac{\langle \mathbf{v}, \mathbf{u} \rangle}{\langle \mathbf{u}, \mathbf{u} \rangle}\mathbf{u},$$

Classical Gram-Schmidt Algorithm

Set up matrix
Initialize Q and R matrices

- Q as mxm square
- R as mxn matrix

%Classical Gram-Schmidt
Loop through each column in matrix

- Convert each column until current into vectors
- Loop through each column vector until current
    - Orthogonalize column as vector
    - Get projection of previous vectors
        - subtract sum from current vector
- Normalized coefficients go into R matrix
- Orthogonalized vector goes into Q

Modified Gram-Schmidt Algorithm

Set up matrix
Initialize Q and R matrices

- Q as mxm square
- R as mxn matrix

%Modified Gram-Schmidt
Initialize v as m x n zero matrix
Loop through each column in matrix
Column goes into matrix
Loop through each column in matrix
Normalized vector until current go into R
Orthogonalized vector goes into Q
Loop through next current column until n
Orthogonalize vectors

Why Modified Algorithm???

Generate Hilbert matrix of size m x n

Forward error: $|| x - x_a || = || A - Q*R ||$

- Classical: $2.7798 * 10^{-14}$
- Modified: $1.2957 * 10^{-16}$

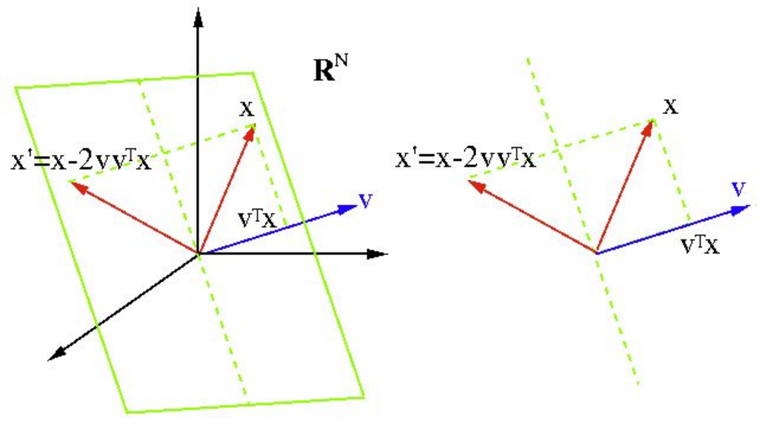
Backward error: $|| f(x) - f(x_a) || = || QQ^T - I ||$

- Classical: 991.9685
- Modified: 1.7776

*Note Machine epsilon constant is 10^{-16}

Applications of QR Factorization

- Not used directly for Image Compression (not good for approximation)
- Least Squares Solution
- Solving system of equations using $Ax = b$
  - $Ax = QRx = b$
  - $x = R^{-1}Q^{T}b$



Singular Value Decomposition (SVD)

Suppose A is the original matrix, then A = USV^T

$$
\underset{m \times n}{\overset{\hat{X}}{\begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & \\ \vdots & \vdots & \ddots & \\ x_{m1} & & & x_{mn} \end{pmatrix}}}
\approx
\underset{m \times r}{\overset{U}{\begin{pmatrix} u_{11} & \cdots & u_{1r} \\ \vdots & \ddots & \\ u_{m1} & & u_{mr} \end{pmatrix}}}
\underset{r \times r}{\overset{S}{\begin{pmatrix} s_{11} & 0 & \cdots \\ 0 & \ddots & \\ \vdots & & s_{rr} \end{pmatrix}}}
\underset{r \times n}{\overset{V^{\mathsf{T}}}{\begin{pmatrix} v_{11} & \cdots & v_{1n} \\ \vdots & \ddots & \\ v_{r1} & & v_{rn} \end{pmatrix}}}
$$

Left singular vectors (Orthogonal matrix)

Singular Values

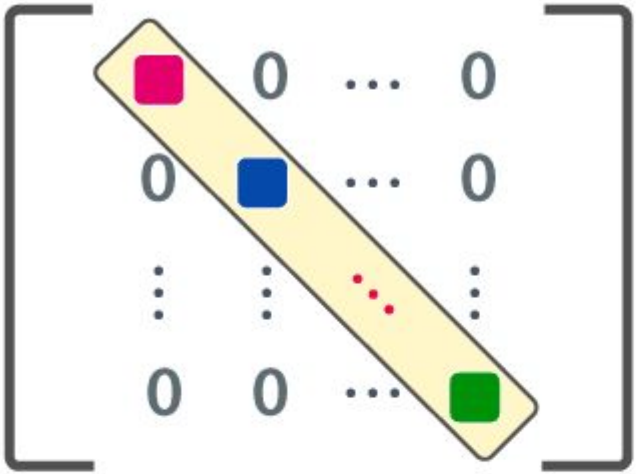
Right singular vectors (Orthogonal matrix)

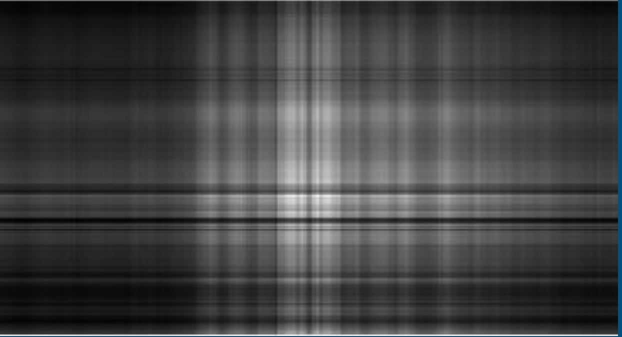
Properties of SVD

- Properties of SVD
  - Rank of matrix is number of non-zero entries in S.
  - Suppose the matrix is an nxn matrix, then determinant of matrix is product of diagonal entries in S.
  - If A is an invertible m x m matrix, then $A^{-1} = VS^{-1}U^T$
  - The m x n matrix can be written as the sum of rank-one matrices (Important!)
- Low-Rank Approximation Property (stated above): The m x n matrix A can be written down as the sum of rank-one matrices = $A = \sum s_i v_i v_i^T$ (from i = 1 to r) where r is the rank of A, and u_i and v_i are the ith columns of U and V, respectively.

Overview of Image Compression

- Type of data compression for digital image as a way to reduce cost for transmission or storage
  - Example: attempt to email pictures through your iPhone (factors depend on storage, wi-fi, speed, etc.)
- Process:
  - Convert image to grayscale image
  - Convert grayscale image to matrix
  - Apply Singular Value Decomposition to matrix
  - Find minimum number of eigenvectors (and thus, singular values) to keep to preserve grayscale image (e.g. zero the last 900, 800, 700, …, 100 or unti get an image the closest to original grayscale image).


0 ⋯ 0
0 ⋯ 0
0 0 ⋯







From top left corner to right
Then go down and go right until
Bottom middle: 40, 140, 240, 340, 440, 840, original picture

Overall results

- It took about 140 singular values (or zeroing last 800 eigenvalues) to get some general replication of the image itself with somewhat clear outlines.
- The closest production of the image took about 440 singular values t(or zeroing last 500 eigenvalues) to get the closest replication of the original image itself

Questions?