Linear Methods for Image Compression

Math 420, Prof. Beezer

Aidan Meacham

University of Puget Sound

Outline

Outline

RGB Color Space

- Intensity and Representation
- Gamut mapping and Translation
- Absolute Color Spaces

Lossy vs. Lossless Methods

- Lossless Methods - GIF / LZW
- Usefulness of Lossy Compression
- Limit - arithmetic, entropy, and LZW coding

Outline

Preliminaries
- Color Spaces
- Lossy vs. Lossless

Methods
- SVD
- PCA
- DCT

SVD

Definition

- A is a matrix with singular values $\sqrt{\sigma_1}, \sqrt{\sigma_2}, \ldots, \sqrt{\sigma_r}$, where r is the rank of A^*A and σ_i are eigenvalues of A
- Define $V = [\mathbf{x}_1|\mathbf{x}_2|\ldots|\mathbf{x}_n]$, $U = [\mathbf{y}_1|\mathbf{y}_2|\ldots|\mathbf{y}_n]$ where $\{\mathbf{x}_i\}$ is an orthonormal basis of eigenvectors for A^*A and $\mathbf{y}_i = \frac{1}{\sqrt{\sigma_i}}A\mathbf{x}_i$

SVD

- Additionally, $s_i = \sqrt{\sigma_i}$
-

$$S = \begin{bmatrix} s_1 & & & & & & \\ & s_2 & & \text{\Large 0} & & & \\ & & \ddots & & & & \\ & & & s_r & & & \\ & \text{\Large 0} & & & 0 & & \\ & & & & & \ddots & \\ & & & & & & 0 | \text{\Large 0} \end{bmatrix}$$

- Thus,

$$AV = US$$
$$A = USV^*$$

SVD Truncated Form

- $A = \sum_{i=1}^{r} s_i \mathbf{x}_i \mathbf{y}_i^*$, where r is the rank of A^*A and the s_i are ordered in decreasing magnitude, $s_1 \geq s_2 \geq \cdots \geq s_r$
- For $i < r$, this neglects the lower weighted singular values
- Discarding unnecessary singular values and the corresponding columns of U and V decreases the amount of storage necessary to reconstruct the image

SVD Example

Sage

- Import image and convert to Sage matrix
- Perform SVD decomposition
- Choose number of singular values and reconstruct

SVD Results

Cameraman, 256 elements

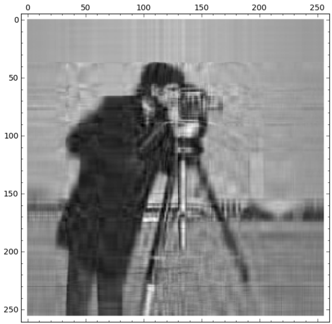
Cameraman, 128 elements

SVD Results

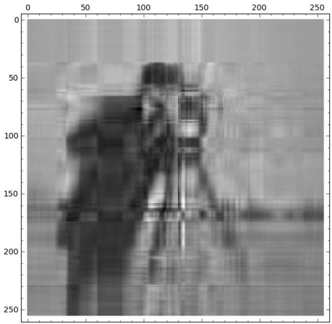
Cameraman, 64 elements

Cameraman, 32 elements

SVD Results



Cameraman, 16 elements



Cameraman, 8 elements

Principal Component Analysis

- Wide variety of applications in many fields:
  - Principal moments and axes of inertia in physics
  - Karhunen-Loeve Transform in signal processing
  - Predictive analytics - customer behavior
- Statistical method for maximizing “variance” of a variable; similar to SVD

Statistics / Information Theory

- Variables with greater variance (higher entropy) carry more information
- Maximizing variance maximizes the information density carried by one variable
- Compress data via approximation, leaving off less significant components
- Weighting - similar to SVD

Statistics / Information Theory

- $E(X) = \sum x_i p(x_i) = \mu$
- “Mean;” average outcome for a given scenario
- $V(X) = E[(X - \mu)^2]$
- Expected deviation from the mean, μ
- Positive square root is standard deviation

Statistics / Information Theory

- Cov(X) or $\sum = E[(X - \boldsymbol{\mu})^T(X - \boldsymbol{\mu})]$
- $\boldsymbol{\mu}$ is the vector of expected values $\mu_i = E(X_i)$
- This matrix is positive semi-definite, which means its eigenvalues will also be positive
- Cov(X) is symmetric, therefore, diagonalizable
- Modal matrix M, composed of rows of eigenvectors for Cov(X), diagonalizes the covariance matrix

Theorem (PCA Finds Principal Axes, via Hoggar[1])

- *Let the orthonormal eigenvectors of Cov(X), where* $X = X_1, \ldots, X_d$*, be* $R_1, \ldots, R_d$
- *Let* X *have components (in the sense of projection)* $\{Y_i\}$*, where* $Y = Y_i$
- *Then* $\{R_i\}$ *is a set of principal axes for* X

Proof.

$$Y_i = X \cdot R_i = XR_i^T$$
$$Y = XM^T, M = \text{Rows}(R_i).$$

Because M diagonalizes $\text{Cov}(X)$, we can write:

$$\text{Cov}(Y) = \text{Cov}(XM^T) = M\text{Cov}(X)M^T,$$

which is a diagonal matrix of eigenvalues; $V(Y_i) = \lambda_i$.
If the R_i are the principal axes for X, then the Y_i will be the uncorrelated principal components, meaning the variance of $X \cdot R_i$ is maximal.
For an arbitrary R, this is only true when$R = R_i$, so $\{R_i\}$ are the principal axes for X. □

PCA Compression

- Given d vectors X, transform into k vectors Y, $k < d$
- Discard Y_{k+1} to Y_d vectors with a minimal loss of data
- Blocks of 8×8 pixels selected; turned into vectors of length $8^2 = 64$
- N vectors stacked as rows into a "class matrix" $H_{N \times 64}$ after subtracting the mean
- Calculate modal matrix, then project data using as many principal components as we like

PCA Remarks

- Less stable than SVD
- Better for *extremely* large data sets
- Big data - consumer modeling

Discrete Cosine Transform

Definition

The one-dimensional DCT can be written as follows, where ϕ_k is a vector with components n, written as a variable to avoid confusion with matrix notation

$$\phi_k(n) = \begin{cases} \sqrt{\frac{2}{N}} cos\frac{(2n+1)k\pi}{2N}, & \text{for } n = 1, 2, \ldots, N-1, \\ \sqrt{\frac{1}{N}}, & \text{for } n = 0. \end{cases}$$

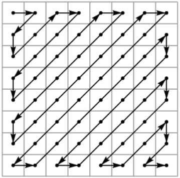
DCT Properties

- A set of k vectors (each of dimension n) is orthonormal
- The matrix of columns $M = [\phi_0|\phi_1| \ldots |\phi_{N-1}]$ is invertible by its transpose
- 2D case: apply transformation first to rows, then to columns (separable; composition of function along each dimension)
- A matrix of values can be transformed via the calculation $B = MAM^T$

JPEG File Format

- JPEG utilizes DCT
- Applying DCT moves information to lower indices (vector or matrix)
- Higher index entries close to zero
- Lossy compression - quantization
- Settings - force the last n indices of a vector to zero
- For every 8×8 submatrix, $(8 - n)^2$ coefficients out of 64 nonzero

JPEG Continued

- The transformed array undergoes zigzag reordering to take advantage of zeroes in the larger indices
- 
- This array is compressed via Huffman encoding (lossless entropy-based algorithm)
- Huffman encoding utilizes a variable-length code table to construct a frequency-sorted binary tree

JPEG Remarks

- Only non-reversible step is quantization
- Reversing other steps (switching order of multiplication) retrieves image
- Data lost no matter what - rounding errors
- DCT transforms frequencies, not intensities - human eye sensitivity / recognition
- Blocky artifacts - natural vs. manufactured images

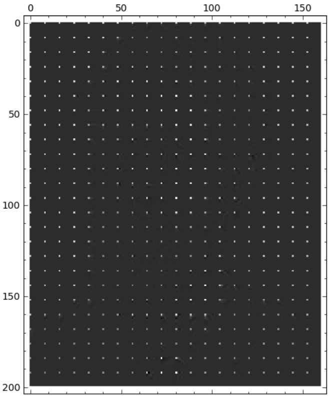
DCT Example

Sage

- Import image and convert to Sage matrix
- Create DCT matrix
- Subdivide matrix and apply transform
- Quantize
- Reconstitute

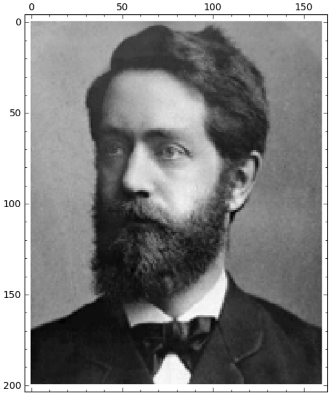
DCT Applied

Klein, 8 elements

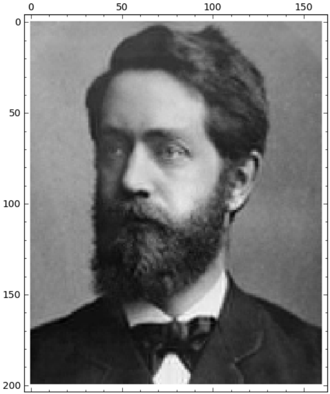


Klein, post-DCT

DCT Results



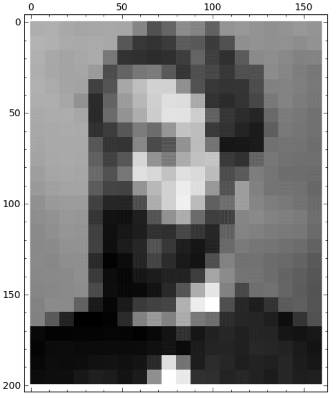
Klein, 8 elements



Klein, 5 elements

DCT Results

Klein, 3 elements



Klein, 1 element

References

- [1] Hoggar, S. G. *Mathematics of digital images: creation, compression, restoration, recognition.* Cambridge: Cambridge University Press, 2006.
- [2] N. Ahmed, T. Natarajan and K.R. Rao. "Discrete Cosine Transform." *IEEE Trans. Computers*, 90-93, Jan. 1974.
- [3] Lay, David. *Linear Algebra and its Applications.* New York: Addison-Wesley, 2000.
- [4] Trefethen, Lloyd N., and David Bau. *Numerical linear algebra.* Philadelphia: Society for Industrial and Applied Mathematics, 1999.