Image Denoising via Low-Rank Approximation and Optimal Hard Thresholding

MAT 167 - Applied Linear Algebra

Koby Taswell and Ayush Garg

University of California, Davis

September 12, 2021

Overview

1. Theory
2. Simple Example - Kingfisher
3. Hard Application - Medical Imaging

Singular Value Decomposition

- Suppose we want to decompose a matrix with a method analogous to Eigenvalue decomposition, but applied to all matricies, not just square matricies [1].
- **Singular value decomposition** provides ability to generalize from $A = \Phi\Lambda\Phi^{-1}$ to $A = U\Sigma V^T$
- For a given matrix A which is size $m \times n$, U is the left unitary matrix forming an orthonormal basis for $\mathbb{R}^m$ while V is the right unitary matrix forming an orthonormal basis for $\mathbb{R}^n$
- Σ is an $m \times n$ diagonal matrix of the singular values where $\sigma_1 \geq \sigma_2 \geq \sigma_n$.
- Each combination of of $\mathbf{u}_i$, σ_i, and $\mathbf{v}_i$ for $1 \leq i \leq \text{rank}(\Sigma)$ creates a rank one matrix $A_i = \sigma_i \mathbf{u}_i \mathbf{v}_i$.
- We can then approximate the original matrix A by performing a **low rank approximation** of rank k with $A_k = \sum_{i=1}^{k} \sigma_i \mathbf{u}_i \mathbf{v}_i$

Low Rank Approximation

- One way to look at singular values is the amount that a specific pair of $\mathbf{u}_i, \mathbf{v}_i$ contributes to the reconstruction of A.
- We know from properties of the SVD, that the number of singular values in Σ is the rank of the matrix.
- Suppose we have a matrix A that is not full rank. A sufficient low rank approximation, would be to **truncate** U and V to only use a number of columns equal to the rank of our matrix. Then $A = \hat{U}\hat{\Sigma}\hat{V}^T = \sum_{i=1}^{r} \sigma_i \mathbf{u}_i \mathbf{v}_i^T$.
- This let's us use less data to store the same or approximately similar information for A.
- So, what if we want to use even less space to store the same or approximately similar information?
- Additionally, what if we have some undesired variation of information called **noise** within the dataset, can we remove this unnecessary data by doing a low rank approximation?

Thresholding

- In essence, we want to find an approximation where if a singular value does not contribute enough to the matrix A, we disregard it.

Note

To simplify notation and reduce confusion we will refer to singular values as y and noise level as σ going forward

- How can we determine this?
- Choose a threshold τ, if a singular value σ_i is below our threshold $(y_i < \tau)$, then set $y_i = 0$
- In doing so, we find an estimation of A which uses less data and is of lower rank. For the values of $y_i = 0$, the corresponding $\mathbf{u}_i \mathbf{v}_i$
- In general we can choose $\tau = \lambda \sqrt{n} \sigma$[2] where n is the size of a square matrix and λ is some coefficient typically between 1 and 10.

Finding our Threshold τ based on λ

- Bulk Edge Thresholding which excludes any singular values below the 'gap' or 'elbow' when plotted. $\lambda = (1 + \sqrt{\beta})$ where $\beta = \frac{m}{n}$.
- Chatterjee proposed that there exists some λ regardless of rank or shape of matrix which would give a near optimal mean square error between the original matrix A and the reconstructed low rank approximation. In his paper, Chatterjee suggested $\lambda \approx 2.02$ [2].
- In 2013, Gavish & Donoho defined two methods:
  1. For an arbitrary $m \times n$ matrix where σ is unknown: $\tau = 2.858 y_{\text{med}}$ and y_{med} is the median of the singular values. We'll refer to this as **Arbitrary Thresholding**
  2. For a square $m \times n$ matrix where σ is known: $\tau = \lambda_\beta \sqrt{n}\sigma$. We'll refer to this as **Non-arbitrary Thresholding**
  - When matrix is $n \times n$, $\beta = 1$ and they define the optimal $\lambda_\beta = \frac{4}{\sqrt{3}} \approx 2.3094$
  - When matrix is $m \times n$, $\beta \neq 1$, use different λ_β see paper for more[3]

Simple Example - Kingfisher

A Noisy Image

- Suppose we have noise in an image. We can simulate this by adding noise to our beautiful picture of a Kingfisher in MATLAB with default `imnoise(I,'gaussian')`
- Image Properties: 1724×1724 in full JPG RGB color scale from 0 to 255 for Red, Green, and Blue. Noise Introduced $r_{red} = 11.1265$dB, $r_{green} = 6.8694$dB, $r_{blue} = 7.0581$dB

Measuring Noise in an Image

- **Mean Squared Error:** Measure the average difference between two matricies by performing MSE $= \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}(a_{ij} - \hat{a}_{ij})^2$
- **Frobenius Norm:** Calculate $\|A - \hat{A}\|_F$ to find the 'distance' between the two matricies[4].
- **Signal-Noise Ratio:** In general SNR $= \frac{P_{signal}}{P_{noise}}$ where P refers to the power of the signal or noise. More specifically this could be SNR $= \frac{s^2}{EN^2}$ where E is the expected value and N is the random noise. Typically measured in decibels on a $\log_{10}$ scale.

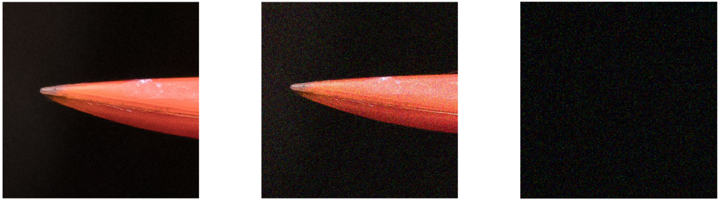


Figure: Comparison of Zoomed in Beak between Original, Noisy, and Noise only Images

Removing Noise with Arbitrary Thresholding

- Methods for Arbitrary Thresholding:
  1. Calculate SVD of Noisy Image for each color channel Red, Green, and Blue
  2. Find y_{med}
  3. Apply threshold $\tau = 2.858 y_{\text{med}}$
  4. Find $\hat{A} = \hat{U}\hat{\Sigma}\hat{V}^T$

Figure: Comparison of Original, Noisy, and Arbitrary Threshold Denoised Images

Removing Noise with Non-arbitrary Thresholding

- Methods for Non-arbitrary Thresholding:
  1. Calculate SVD of Noisy Image for each color channel Red, Green, and Blue
  2. Find signal to noise ratio σ
  3. Since the image is $n \times n$, Apply threshold $\tau = \frac{4}{\sqrt{3}}\sqrt{n}\sigma$
  4. Find $\hat{A} = \hat{U}\hat{\Sigma}\hat{V}^T$

Figure: Comparison of Original, Noisy, and Non-arbitrary Threshold Denoised Images

Removing Noise with Bulk Edge Thresholding

- Methods for Bulk Edge Thresholding:
  1. Calculate SVD of Noisy Image for each color channel Red, Green, and Blue
  2. Find $\beta = \frac{m}{n}$
  3. Apply threshold $\tau = (1 + \sqrt{\beta})\sqrt{n}\sigma$
  4. Find $\hat{A} = \hat{U}\hat{\Sigma}\hat{V}^T$

Figure: Comparison of Original, Noisy, and Bulk Edge Threshold Denoised Images

Performance

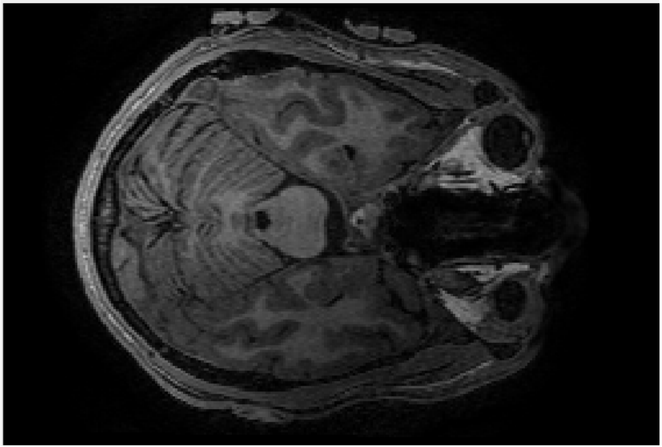
- A keen eye might notice that there does **not** appear to be a large amount of difference between the results of our three thresholding methods.
- By evaluating original MSE of the raw image in comparison to the noisy image we find $\mathrm{MSE}(I, I_{noisy}) = 0.0066$, a small but not insignificant amount of variance.
- When calculating the MSE of the raw image in comparison to all three of the denoised images, we find the exact same value $\mathrm{MSE}(I, I_{denoised}) = 0.0022$ meaning that approximately 2/3rds of the variance or noise in the image remained.
- However none of the three methods, despite having slightly different threshold values, had significant difference in results.
- Another important note is that in losing a lot of noise, we've also lost some amount of signal (See pixelation on the branch and elsewhere). This is less noticeable since our image is already high resolution aka the matrix is large but will be easily noticeable in images that begin with significantly lower resolution. This appears in our $\sigma = \mathrm{SNR}$ as higher noise ratio than originally measured, however the MSE remains low.

Results

Method	MSE	Frobenius	Signal to Noise (Db)	Approximation Rank k
Arbitrary	0.0022	79.963	$r_{red} = 15.5806$ $r_{green} = 12.3801$ $r_{blue} = 11.7773$	$k_{red} = 54$ $k_{green} = 45$ $k_{blue} = 56$
Non-Arbitrary	0.0022	80.781	$r_{red} = 15.3444$ $r_{green} = 12.3539$ $r_{blue} = 11.7583$	$k_{red} = 33$ $k_{green} = 41$ $k_{blue} = 41$
Bulk Edge	0.0022	81.282	$r_{red} = 15.5387$ $r_{green} = 12.1211$ $r_{blue} = 11.6600$	$k_{red} = 44$ $k_{green} = 71$ $k_{blue} = 75$

Table: Comparison of results when denoising and approximating $r = 1724$ Red, Green, and Blue color matricies

Hard Application - Medical Imaging



Medical Imaging Data

- One very common application of denoising is to the very complicated issue of medical imaging data.
- Similar to how photography and cameras have immensely improved over the decades, medical imaging technology for Positron Emission Tomography (PET), Magnetic Resonance Imaging (MRI), and more have immensely improved
- Unlike cameras and typical photography, medical imaging aims to capture 3D volume so we can better understand the anatomy or function of a region of the body.
- This 3D volume is acquired by taking a series of pictures at different positions through the volume, then stacking them together. Different imaging techniques use different methods to find the picture at each position, however, in all cases the pictures come together to create a 3D volume.
- Also unlike cameras, although huge improvements have been made, noise from the imaging techniques can make it very hard to understand certain types of data.

Denoising Medical Data

- Since medical data is 3D, not 2D, it is a lot more challenging then merely applying some of the denoising techniques discussed previously.
- Additionally, medical data is not as high resolution as common photographs and may be described by a much more sparse matrix, causing low rank approximation to have much more error.
- Many different high level complex denoising techniques exist and work very well.
- In this presentation, we will instead take simplistic approach by attempting to only apply the basic techniques previously discussed at each 2D image slice in the overall volume.
- *Disclaimer/Spoiler:* It is completely nonsensical to denoise a volume by each 2D image since noise can be distributed throughout the volume. This presentation will display a situation where using these basic techniques are **not** optimal.

A Noisy Dataset

- Suppose we take a brain scan volume from the Autism Brain Imaging Data Exchange (ABIDE) dataset
- Input data size was in the form of $172 \times 256 \times 256$ of integer values representing intensity values of MRI signal.
- Instead of trying to uniformly add noise across the entire volume, we'll add noise and attempt to denoise each slice of the brain scan along the 1st dimension, giving us 172 square image matricies, each of size 256×256

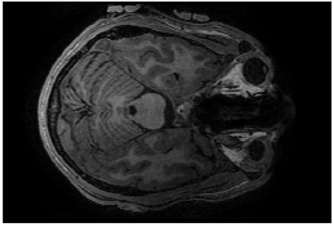
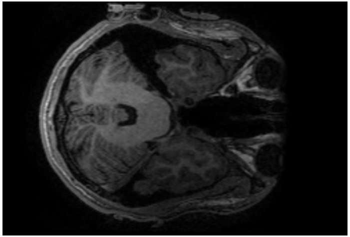


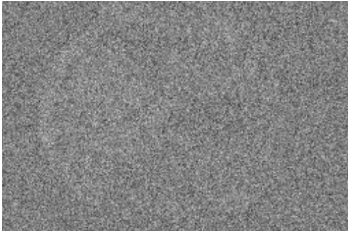
Figure: A single image slice from the center of the dataset

Removing noise with Thresholding

- For each image slice in the 1st dimension:
  1. Add noise with default `imnoise(I,'gaussian')`
  2. Denoise with arbitrary thresholding $\tau = 2.858 y_{\text{med}}$
  3. Find $\hat{A} = \hat{U}\hat{\Sigma}\hat{V}^T$
  4. Measure the signal to noise ratio for both the noisy and denoised image.
- The process was also repeated with non-arbitrary thresholding



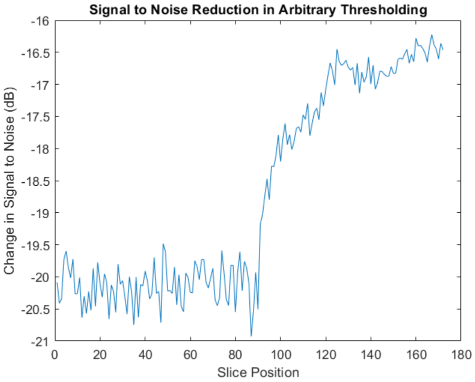
(a) Original image of middle of head

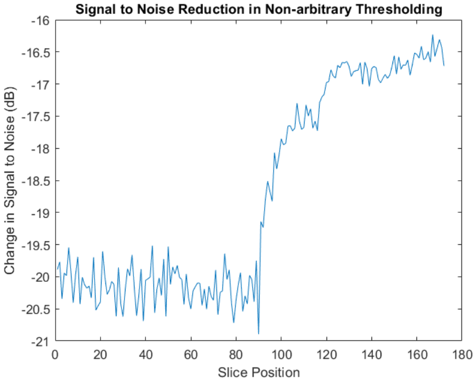


(b) Image with noise added
Perhaps added way too much noise

(c) Failure of Reconstruction

Results



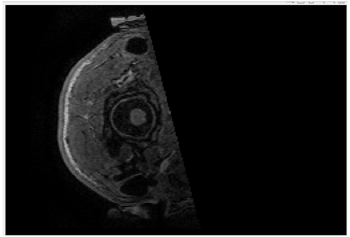


- Yes those numbers are negative!

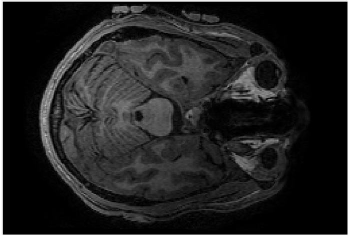
Discussion

- What do negative $\Delta\sigma$ values mean?
  - Our reconstructed image took out some noise, but lost so much signal that the difference appears both numerically and visually as even more noise than before.
  - Average noise per image after adding noise was 14.07dB, almost 1000 times more noise added than noise added to red matrix of kingfisher
  - Negative values this high mean that the image is no longer recognizable thus rank of approximation is too low. This is easily verifiable by visual inspection.
- We see an interesting change with higher negative values in slices 0-100, then a jump from $\Delta\sigma \approx -20$ up to $\Delta\sigma \approx -16.5$. Why?
  - Let's take a look at the volume to try to figure this out.

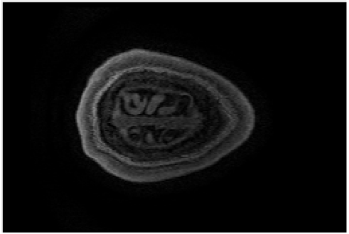
Some Observations



(a) Image of lower-middle of head, sheared to protect identity



(b) Full image of middle of brain



(c) Sparse image at top of the brain

- Clean continuous regions (middle of brain and higher) perform slightly better due to no abrupt shearing and loss of data
- Images around slice 90 and below were all sheared so had non-continuous regions.

Conclusions

- On a large matrix such as the kingfisher picture, low rank approximation allows us to recreate an image with low mean-square error, thus high similarity.
- Loss of signal appears as noise in signal-to-noise metrics, however, we can verify that it is loss of signal, not added noise since MSE is low. Reducing the overall resolution would likely reduce the impact this has visually on an image.
- On a smaller matrix such as brain imaging data, these methods are **not** accurate with too much loss of signal.
- Further investigation would include a comparison of hard thresholding to other methods such as soft thresholding, wavelet transform domain methods[5], convolutional neural network based methods, and more [6]

TL:DR - Too Long, Didn't Read

- Although there may exist some *optimal thresholding coefficient* for hard thresholding, Hard thresholding is *not always the optimal method*.
- Choose the right tool for the right job and don't assume a method will work well for all input data of similar types.

References

[1] C. Eckart and G. Young, "The approximation of one matrix by another of lower rank," *Psychometrika*, vol. 1, no. 3, pp. 211–218, Sep. 1936. DOI: 10.1007/bf02288367.

[2] S. Chatterjee, "Matrix estimation by universal singular value thresholding," *The Annals of Statistics*, vol. 43, no. 1, Feb. 2015. DOI: 10.1214/14-aos1272.

[3] M. Gavish and D. L. Donoho, "The optimal hard threshold for singular values is 4/sqrt(3),", May 2013. arXiv: 1305.5870 [stat.ME].

[4] A. A. Shabalin and A. B. Nobel, "Reconstruction of a low-rank matrix in the presence of gaussian noise," *Journal of Multivariate Analysis*, vol. 118, pp. 67–76, Jul. 2013. DOI: 10.1016/j.jmva.2013.03.005.

[5] C. Taswell, "The what, how, and why of wavelet shrinkage denoising," *Computing in Science & Engineering*, vol. 2, no. 3, pp. 12–19, 2000. DOI: 10.1109/5992.841791.

[6] L. Fan, F. Zhang, H. Fan, *et al.*, "Brief review of image denoising techniques," *Visual Computing for Industry, Biomedicine, and Art*, vol. 2, no. 1, Jul. 2019. DOI: 10.1186/s42492-019-0016-7.