Extended Yale Faces B Database – Eigenfaces

Nick Moore

January 23, 2017

Abstract

Decompositions are a useful method for generating simplified matrices from a complex matrix. One common method is singular value decomposition (SVD). Herein SVD is applied to a matrix M composed of 2414 different images of human faces from the Yale Face Database study. SVD of a matrix M results in M = UΣV*, with each resultant matrix containing eigenvectors and singular values from M. Matrices U, Σ, and V are analyzed to determine their representation of the original image matrix M, and images are reconstructed to establish what rank is required for accurate reconstruction.

Sec. I. Introduction & Overview

The goal of this report is to explore the singular value decomposition method and its results in order to better understand how they work for a real data set. Cropped images from the Extended Yale Face Database B were used for the analysis.[1] Images were first altered to form column vectors and then placed in matrix M, where each column is a unique image. M was decomposed through SVD, which generated three new matrices. The generated matrices U, Σ, and V were analyzed to determine what aspect of the image data each matrix contains, the relative importance of the eigenvectors and singular values, and how much of this information is required for image reconstructions. The methods and information used in this work can be applied to other data sets because SVD is able to work on any matrix. Matlab 2015b was used for all of the work presented. The functions used and the entire code can be found in Appendices A & B respectively.

Sec. II. Theoretical Background

Singular value decomposition (SVD) is a method of matrix decomposition that breaks down a matrix M into three new, simpler matrices U, Σ, and V. Equation 1 below shows how an *m x n* matrix M relates to U, Σ, and V.[2] These new matrices can be thought of as scaling or rotating matrices that when combined in equation 1 give back the original matrix. U and V are rotation matrices of eigenvectors where. U is an *m x n* unitary matrix and V is an *n x n* unitary matrix. It is important to note that when using equation 1 the conjugate transpose of V is used. Σ is the scaling matrix of the square root of eigenvalues, also known as singular values. It is a diagonal matrix of *m x n* size. To calculate the SVD of a matrix M equations 2 and 3 are used.[2]

$$M = U\Sigma V^* \qquad \text{Equation 1}^2$$

$$\begin{aligned} M^T M &= (U\Sigma V^*)^T (U\Sigma V^*) \\ M^T M &= V\Sigma U^* U\Sigma V^* \\ M^T M &= V\Sigma^2 V^* \\ M^T M V &= V\Sigma^2 \end{aligned} \qquad \text{Equation 2}^2$$

$$\begin{aligned} MM^T &= (U\Sigma V^*)(U\Sigma V^*) \\ MM^T &= U\Sigma V^* V\Sigma U^* \\ MM^T &= U\Sigma^2 U^* \\ MM^T U &= U\Sigma^2 \end{aligned} \qquad \text{Equation 3}^2$$

Figure 1: Singular Values

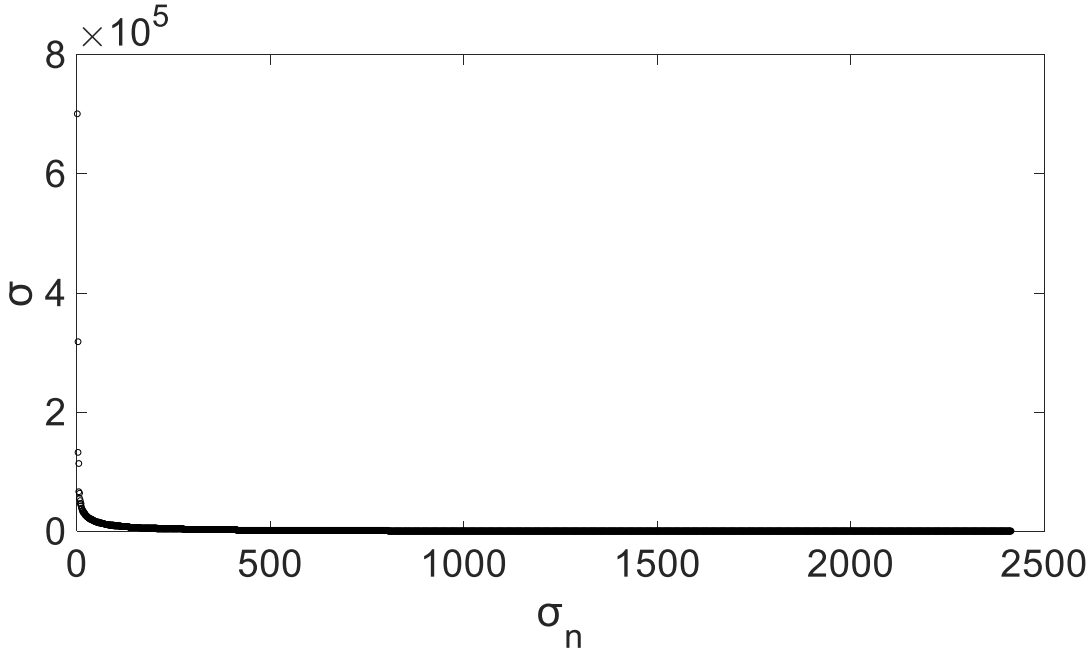

Figure 1: Plot of singular values σ_1 to σ_{2414}.

From here the eigenvectors and eigenvalues can be calculated to give matrices U and V (eigenvectors) and Σ^2 (eigenvalues). Singular values are equal to the square root of the eigenvalues.

Sec. III. Algorithm Implementation & Development

In order to analyze the cropped images from the Yale database they first needed to be reshaped and placed in a matrix. Every image was converted to a column vector and added to matrix M as a new column, so once completed matrix M had a column from each of the imported images (Appendix B lines 13-19).[3] SVD was performed on matrix M and returned in matrices U, S, and V, where S corresponds to Σ (Appendix B line 21). Next the singular values can be plotted to determine the importance of each in image reconstruction. Figure 1 shows a plot of the singular values and figure 2 shows the log plot of the singular values both plotted against σ_n which ranges from σ_1 to σ_{2414} (Appendix B lines 23-27). The next step was to explore image reconstruction with varying ranks. Rank corresponds to the singular values, so a rank of 5 includes σ_1 through σ_5. Original images were reconstructed using equation 1, but matrix S was modified to include a limited number of ranks. The image reconstructions are shown in figure 3. Ranks of 1, 5, 10, 50, 100, 500, 1000, 1500, 2000, and 2414 (all ranks) were used for image reconstruction (Appendix B lines 29-91). The clarity and detail of the reconstructed image increases with rank because more singular values are being used. The full rank image is identical to the original image because it includes all the information from all three of the matrices. The final plot, Figure 4, shows the first six modes, which are the first four columns of matrix U (Appendix B lines 93-99). The more green an area the higher its value, and the more blue an area the lower its value.

Figure 2: Log of Singular Values

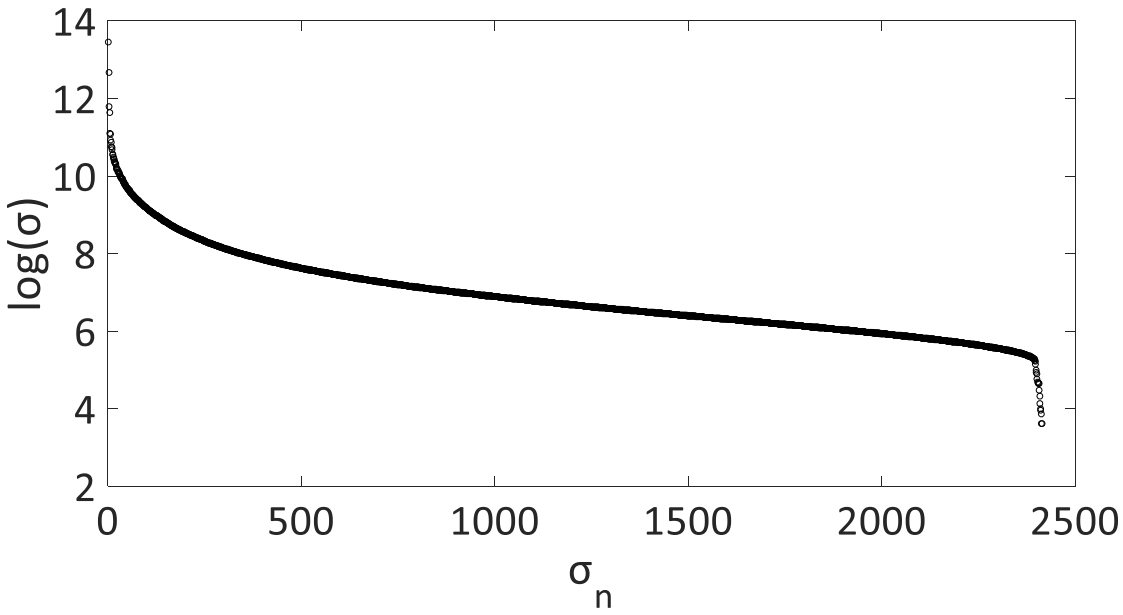

Figure 2: Plot of the log(σ_n) from σ_1 to σ_{2414}

Sec. IV. Computational Results

The previous section detailed how the images were manipulated and underwent SVD to generate figures 1-4. With matrices U, S, and V calculated, using the svd function in MATLAB, the role of these matrices in SVD and usefulness for application can be evaluated. The data used is images of faces, so the matrices will be discussed with this is mind. The eigenvector matrices U and V are basis sets for all of the images used to generate M. U is the face-space, which can be thought of as a coordinate plane for the faces. In addition each column of U is a mode. The modes capture the common features of the images in M. For example the first column of U is the first mode and contains the features shared most frequently by the images. As the column number increases the features become less common. Mode 1 in figure 4 shows an image that looks to be a generic, human face. The image has the defining features of a human face, such as eyes, nose, and mouth, but it lacks specific details that are seen in the imported images. This is because the first mode captures the most important features across all of the images, which are the features that are present across all of the faces. As the modes increase the images become more detailed because they now account for the less common features, such as mouth shape and eyebrows. Mode 4 shows this well because the areas around the mouth, nose, and eyes are beginning to show more detail.

The other eigenvector matrix (V) is also a basis set and it contains information about how the different images are projected onto the face-space (U). V tells how much influence each of the modes has on the reconstructed image. Using facial features as an example, an image with facial hair will have a larger value for its V column vector that corresponds to the mode that accounts for facial hair (with the generalization that facial hair is able to be captured by 1 mode). In figure 3 the same image is shown repeatedly because the same column of V was used. If a different column of V had been selected, then a different image would be recreated. U is the face-space, or coordinate system that we map the images, but V contains the information about how the different modes of U come together to form the unique images.

Figure 3: Image Reconstruction by Rank

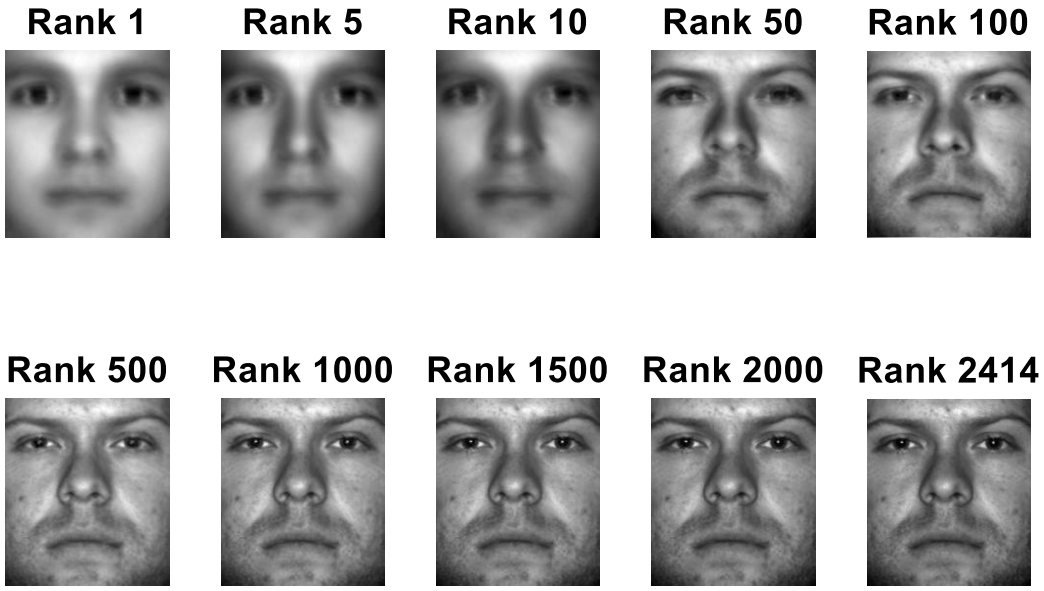

Figure 3: Image reconstructions with different ranks. The rank 2414 image is reconstructed with all ranks.

Σ is the singular value matrix. It is a diagonal matrix that contains all of the singular values (σ) in order from greatest to least ($\sigma_1 > \sigma_2 > \ldots > \sigma_n$). Singular values are a quantification of the importance of the modes from matrix U. Larger σ_n values correspond to modes that capture the most common features of the images. Figure 1 shows a plot of the singular values, which shows how quickly the values drop in an order of magnitude over the first 5 singular values, from $\sigma_1 = 7.0 \times 10^5$ to $\sigma_5 = 6.6 \times 10^4$. Figure 2 also shows how quickly the singular values drop in magnitude, leveling out at around σ_{500}.

Figure 4: Mode Images

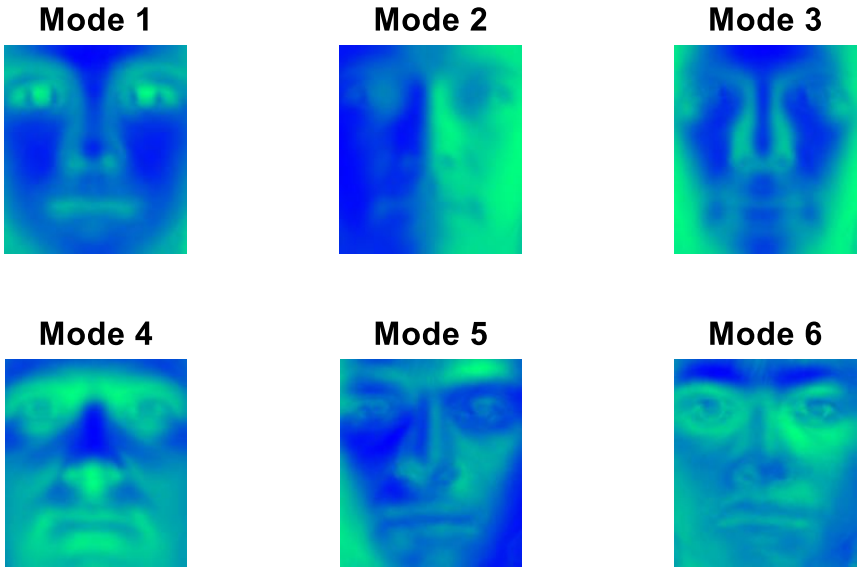

Figure 4: Images of the first 6 modes (columns of U). Green corresponds to higher values, and blue to lower values.

Another method to look at the singular values is through recreated images with different ranks, shown in figure 3. The different ranks of the image correspond to how many singular values are included in the Σ matrix. So a rank of 1 only includes σ_1, while a rank of 100 includes σ_1 to σ_{100}. Higher rank reconstructions result in clearer, more detailed images because they are able to include the higher modes that contain the more detailed information. Based on the images shown a rank of 500 carries enough information for "good" image reconstruction. The ranks higher than 500 produce more detailed image reconstructions, but there is a much larger difference in the images with lower ranks. A rank of 500 relates well with the decrease in angle of the log(σ) plot. The singular values directly relate to image quality as expected.

Sec. V. Summary & Conclusions

Singular value decomposition was applied to the cropped images from the Extended Yale Face Database B and the results were analyzed for the significance and role of the resultant matrices U, Σ, and V where M = UΣV*. Matrices U and V are basis sets for M, and Σ is the diagonal singular value matrix. U contains information for the face-space, which remaps the other matrices into this space. Σ quantifies the modes given in U and is equal to the square root of the eigenvalues. The column vectors of V are used to "tell" how the different images are mapped in the face-space, U. Figure 4 shows qualitatively what aspects of the face are captured by the first six modes (columns) of U, with the most common facial features being found in the lowest modes. The image reconstructions shown in figure 3 and the σ plots in figures 1 and 2 all show that the singular values drop rapidly, and at a rank between 100 and 500 is the minimum for "good" image reconstruction. The processes used here can be expanded upon to develop facial recognition software, showing the wide variety of applications for SVD in the real world.

References

(1) Georghiades, A. S.; Belhumeur, P. N.; Kriegman, D. J. *IEEE Trans. Pattern Anal. Mach. Intell.* **2001**, *23* (6), 643–660.
(2) Kutz, J. N. *Data-Driven Modeling & Scientific Computation: Methods for Complex Systems & Big Data*; Oxford University Press, Inc.: New York, NY, USA, **2013**.
(3) Matlab FAQ. http://matlab.wikia.com/wiki/FAQ#How_can_I_process_a_sequence_of_files.3F

Appendix A: MATLAB Functions

[U,S,V] = svd(M); – Computes the SVD matrix M and assigns the matrices U, Σ, and V to U, S, and V, respectively.

M = reshape(M, [size1 size 2]) – Reshapes matrix M to a new size.

imshow(M) – Generates an image from matrix M.

subplot(m,n,j) – Creates a space for m by n plots or images to be generated at a position j.

Appendix B: MATLAB Code

```
% Specify the folder where the files live.
myFolder = 'X:\Nick Moore\cropped images';
% Check to make sure that folder actually exists.  Warn user if it doesn't.
if ~isdir(myFolder)
    errorMessage = sprintf('Error: The following folder does not exist:\n%s', myFolder);
    uiwait(warndlg(errorMessage));
    return;
end
% Get a list of all files in the folder with the desired file name pattern.
filePattern = fullfile(myFolder, '*.pgm');
theFiles = dir(filePattern);
M = zeros(32256,2414);
for k = 1 : length(theFiles)
    baseFileName = theFiles(k).name;
    fullFileName = fullfile(myFolder, baseFileName);
    imageArray = imread(fullFileName);
    imageVector = imageArray(:);
    M(:,k) = imageVector;
end
[U,S,V] = svd(M);
 %Sigma Matrix Plots
S_diag = diag(S);
figure; plot(S_diag,'ko');
 S_diag_log = log(S_diag);
figure; plot(S_diag_log,'ko');
%Plot the same image at different rank
%values(1,5,10,50,100,500,1000,1500,2000,2414)
%Rank 1 approximation
S_rank1 = S;
S_rank1(2:end,2:end) = 0;
A_rank1 = U*S_rank1*V';
%Rank 5 approximation
S_rank5 = S;
S_rank5(6:end,6:end) = 0;
A_rank5 = U*S_rank5*V';
%Rank 10 approximation
S_rank10 = S;
S_rank10(11:end,11:end) = 0;
A_rank10 = U*S_rank10*V';
%Rank 50 approximation
S_rank50 = S;
S_rank50(51:end,51:end) = 0;
A_rank50 = U*S_rank50*V';
%Rank 100 approximation
S_rank100 = S;
S_rank100(101:end,101:end) = 0;
```

```
A_rank100 = U*S_rank100*V';
%Rank 500 approximation
S_rank500 = S;
S_rank500(501:end,501:end) = 0;
A_rank500 = U*S_rank500*V';
%Rank 1000 approximation
S_rank1000 = S;
S_rank1000(1001:end,1001:end) = 0;
A_rank1000 = U*S_rank1000*V';
%Rank 1500 approximation
S_rank1500 = S;
S_rank1500(1501:end,1501:end) = 0;
A_rank1500 = U*S_rank1500*V';
%Rank 2000 approximation
S_rank2000 = S;
S_rank2000(2001:end,2001:end) = 0;
A_rank2000 = U*S_rank2000*V';
%Rank 2414 approximation
S_rank2414 = S;
A_rank2414 = U*S_rank2414*V';
%Plot the image reconstructions
subplot (2,5,1); imshow(reshape(A_rank1(:,1),[192 168]),[]); title('Rank 1');
subplot (2,5,2); imshow(reshape(A_rank5(:,1),[192 168]),[]); title('Rank 5');
subplot (2,5,3); imshow(reshape(A_rank10(:,1),[192 168]),[]); title('Rank 10');
subplot (2,5,4); imshow(reshape(A_rank50(:,1),[192 168]),[]); title('Rank 50');
subplot (2,5,5); imshow(reshape(A_rank100(:,1),[192 168]),[]); title('Rank 100');
subplot (2,5,6); imshow(reshape(A_rank500(:,1),[192 168]),[]); title('Rank 500');
subplot (2,5,7); imshow(reshape(A_rank1000(:,1),[192 168]),[]); title('Rank 1000');
subplot (2,5,8); imshow(reshape(A_rank1500(:,1),[192 168]),[]); title('Rank 1500');
subplot (2,5,9); imshow(reshape(A_rank2000(:,1),[192 168]),[]); title('Rank 2000');
subplot (2,5,10); imshow(reshape(A_rank2414(:,1),[192 168]),[]); title('Rank 2414');
%Plot of modes
for j=1:6
    subplot (2,3,j);
    ef = reshape(U(:,j),192,168);
    imshow(ef,[]), axis off, shading interp, colormap(winter);
    Casenum = int2str(j);
    title(['Rank',Casenum],'FontSize',35);
end
```