Computer Homework 2

Matrix and Tensor Decompositions

Dr. Sepideh Hajipour

Sharif University of Technology

Electrical Engineering

Nikoo Moradi

400101934

Date: May 17, 2024

Contents

1 Grayscale Image Compression Using SVD

A grayscale image of size 256x256 (e.g., `cameraman.tif` in MATLAB) is selected, and Singular Value Decomposition (SVD) is applied to the image matrix. The steps are performed once with normalized data and once with raw data. My MATLAB code for the analysis, includes loading the image, normalizing the data, performing SVD, reconstructing the image, calculating the RMSE, and plotting the results, once for normalized data and once for raw data.

1.A Methodology

1.A.a Normalization

- Normalize the image data.
- Perform SVD and reconstruct the image for various ranks.
- Calculate and plot the root mean squared error (RMSE).

1.A.b Without Normalization

- Use raw image data.
- Perform SVD and reconstruct the image for various ranks.
- Calculate and plot the RMSE.

1.B Results

1.B.a Normalization

1. **Reconstruct the Image with Different Ranks:**

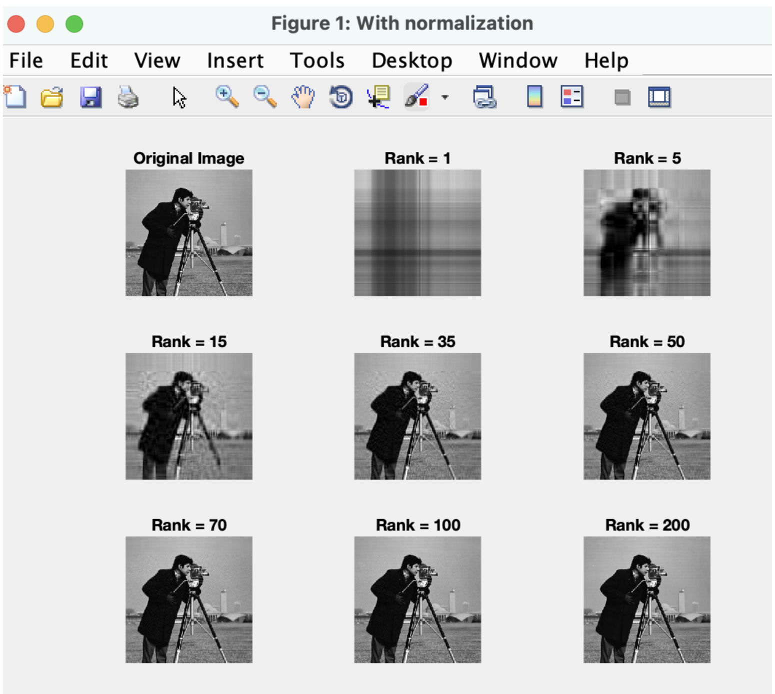


Figure 1: Reconstructed Images with Normalization

2. **Plot the Error vs. Rank:**

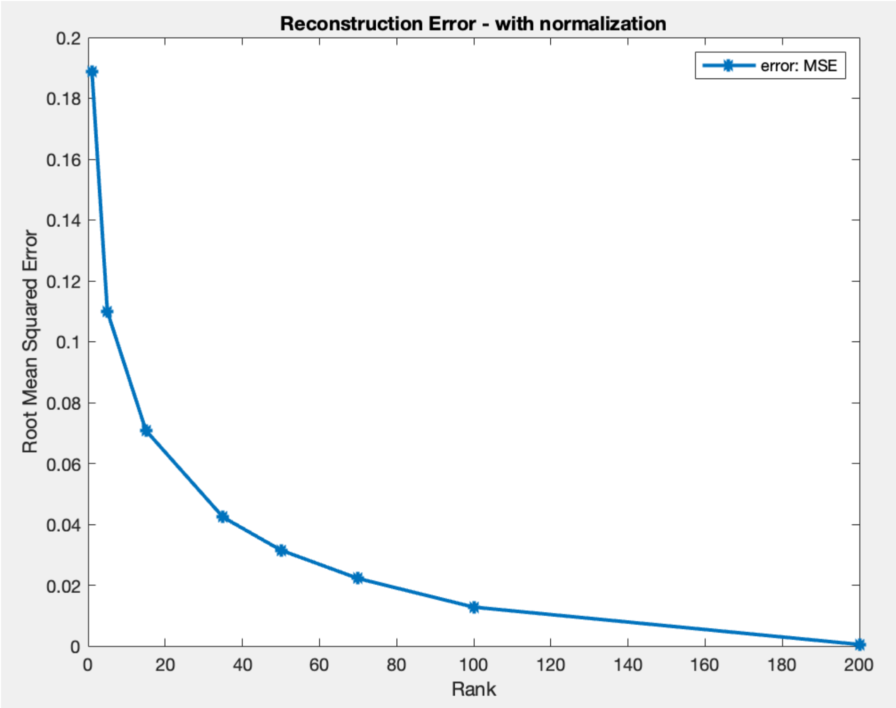


Figure 2: Error Plot vs Rank with Normalization

3. **Plot the Error vs. Compression Ratio:**

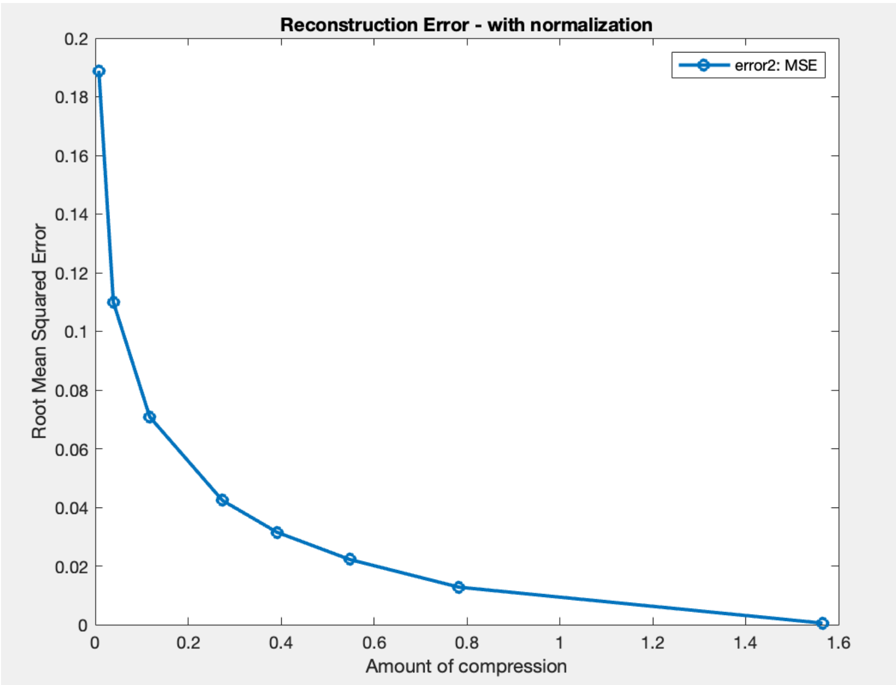


Figure 3: Error Plot vs Compression with Normalization

1.B.b Without Normalization

1. **Reconstruct the Image with Different Ranks:**

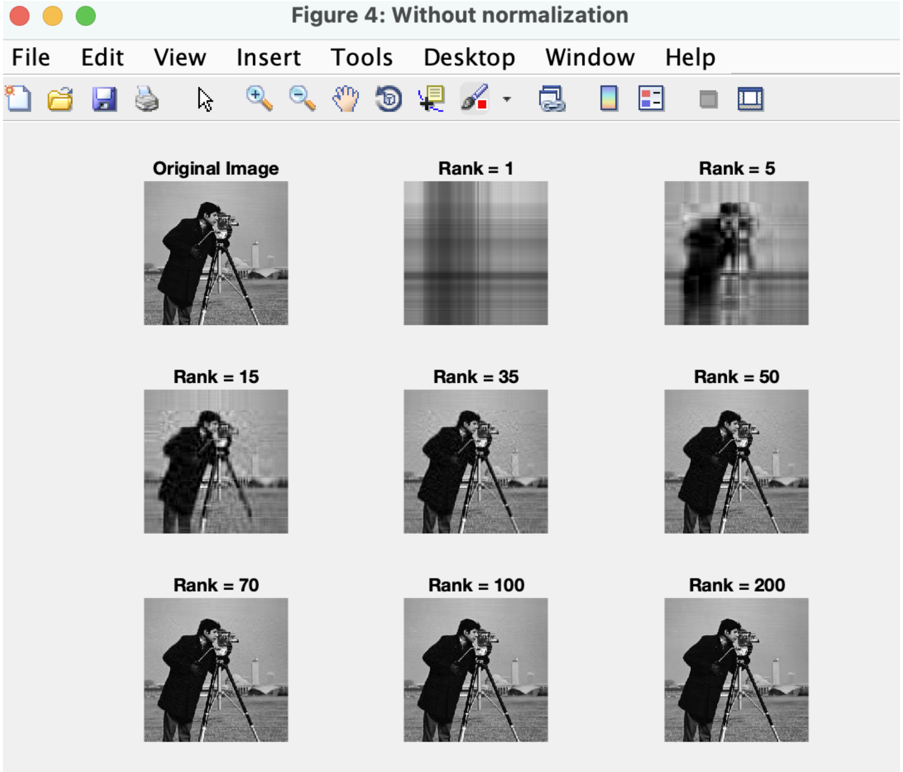


Figure 4: Reconstructed Images without Normalization

2. **Plot the Error vs. Rank:**

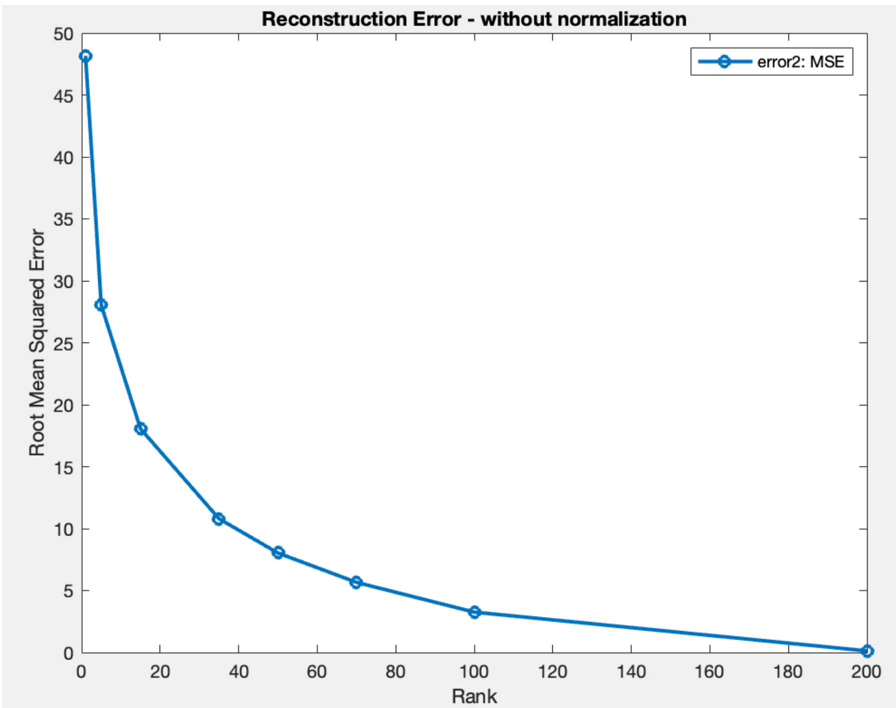


Figure 5: Error Plot vs Rank without Normalization

3. **Plot the Error vs. Compression Ratio:**

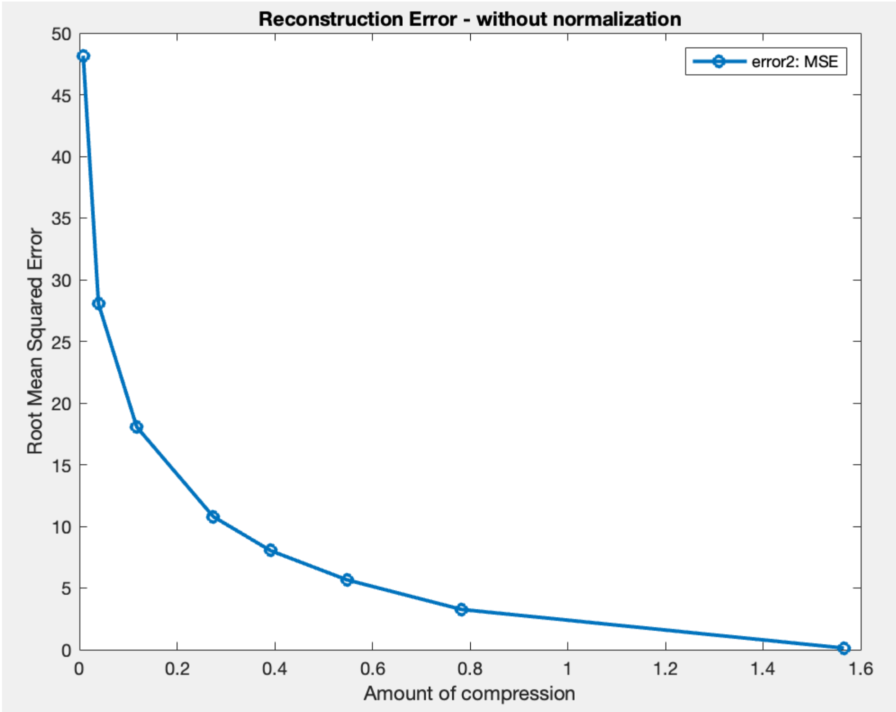


Figure 6: Error Plot vs Compression without Normalization

1.C Conclusion

- **Normalization Impact:** Normalization significantly reduces the RMSE, indicating a better reconstruction quality.
- **Error Analysis:** The RMSE decreases as the rank and compression ratios increase.

2 EEG Signal Denoising Using SVD

The dataset `EEGdata.mat` contains simulated EEG signals in two conditions: noisy (X_{noise}) and noise-free (X_{org}). There are 32 channels and 5000 time samples. The goal is to apply Singular Value Decomposition (SVD) to the noisy signal, remove noise sources, and reconstruct the noise-free signal. The reconstruction error is calculated by comparing the reconstructed signal with the original noise-free signal. The number of non-noise sources is chosen to minimize this error.

2.A Methodology

1. **Load Data:** The EEG data is loaded from the `EEGdata.mat` file.
2. **Perform SVD:** SVD is applied to the noisy signal X_{noise} to obtain matrices U, S, and V.
3. **Reconstruction and Error Calculation:**
   - For each rank k, reconstruct the signal using the first k components.
   - Calculate the reconstruction error using the Frobenius norm of the difference between X_{org} and the reconstructed signal.
4. **Determine Optimal Rank:** Identify the rank k that minimizes the reconstruction error.

2.B MATLAB Code

The MATLAB code for the analysis is provided below. It includes loading the data, performing SVD, reconstructing the signal, calculating the reconstruction error, and determining the optimal number of non-noise sources.

```
EEGdata = load('EEGdata.mat');
Xorg = EEGdata.Xorg;
Xnoise = EEGdata.Xnoise;

[U,S,V] = svd(Xnoise);
disp(size(S));
ranks = 1:1:32;
errors = zeros(size(ranks));

for i = 1:length(ranks)
    k = ranks(i);

    % Reconstruct the signal with k rank
    reconstructed_X = U(:,1:k) * S(1:k,1:k) * V(:,1:k)';

    % Calculate error (Frobenius norm)
    errors(i) = norm(Xorg - reconstructed_X, 'fro');
end

[min_error, min_source] = min(errors);
```

2.C Results

1. **Error Plot vs. Rank:** The plot of reconstruction error versus rank shows how the error changes with different numbers of non-noise sources.

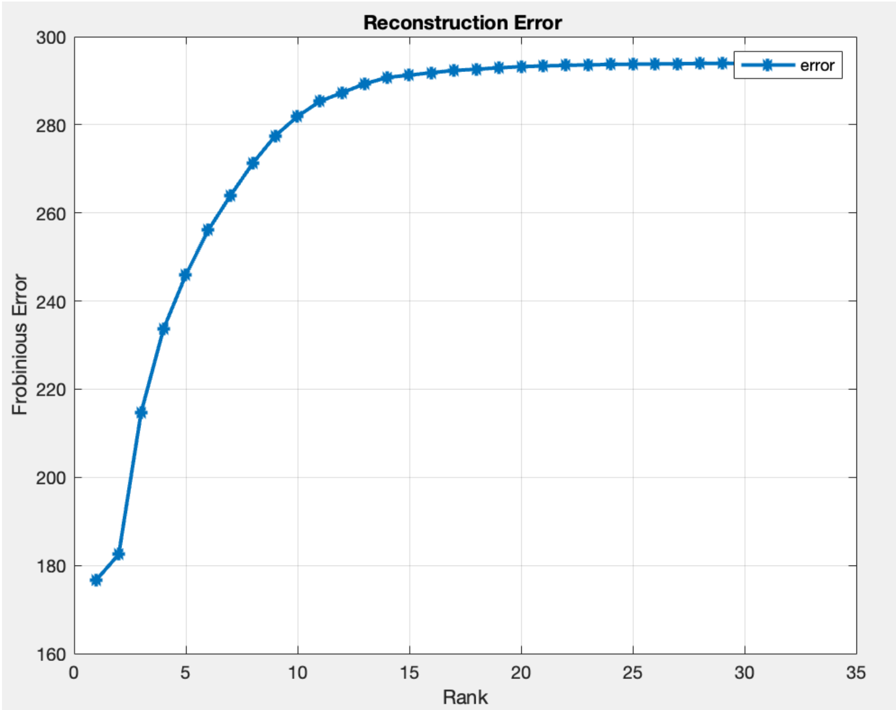


Figure 7: Reconstruction Error Plot

2. **Minimum Error and Optimal Number of Non-Noise Sources:**
   - **Minimum Error:** 176.7286
   - **Optimal Number of Non-Noise Sources:** 1

2.D Conclusion

By applying SVD to the noisy EEG signal and reconstructing it with different ranks, we found that using the first principal component (rank 1) results in the minimum reconstruction error. This indicates that the majority of the signal's energy is captured by the first component, and higher components contribute more to noise than to the actual signal.

3 3D Data Analysis Using SVD and PCA

The dataset `PCAdata.mat` contains 3D observations. This report visualizes the 3D data, examines data elongation in different directions, and uses Singular Value Decomposition (SVD) and Principal Component Analysis (PCA) to extract important statistical insights.

3.A Data Visualization and Analysis

3.A.a Part (a): 3D Scatter Plot

A 3D scatter plot is created using the `scatter3` function to visualize the data distribution and elongation in different directions.

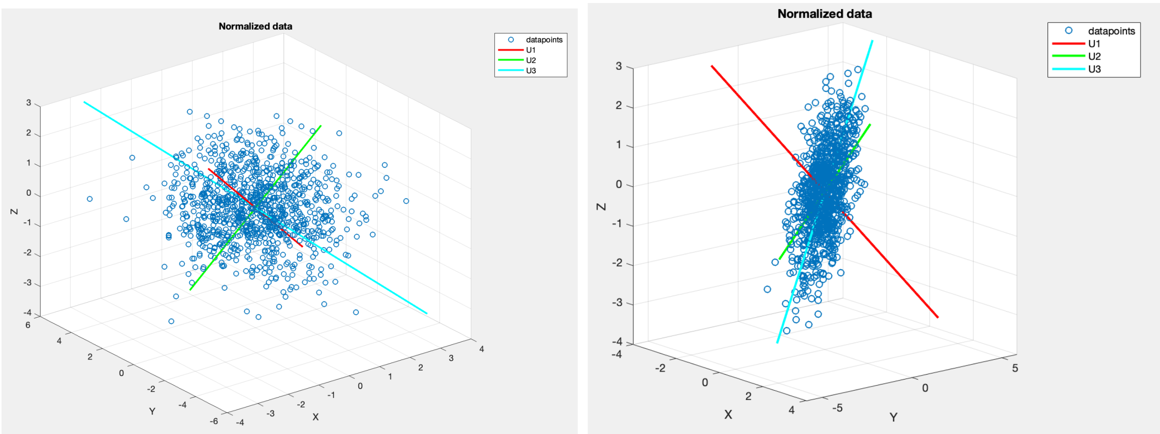
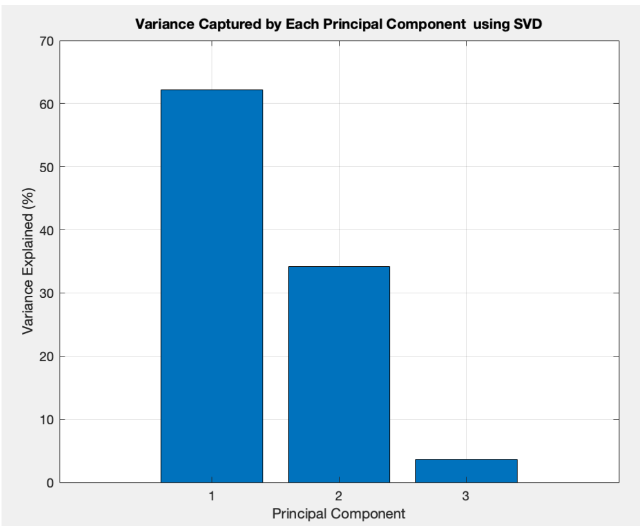


Figure 8: 3D Scatter Plot and New Directions

3.A.b Part (b): SVD Analysis

1. **SVD Computation:** SVD is applied to the normalized data to get the left singular vectors (U), singular values (S), and right singular vectors (V).
2. **Explained Variance:** The variance explained by each principal component is calculated and plotted.




```
data_variance_explained_svd =

   62.1826
   34.1765
    3.6409
```

Figure 9: Variance Explained by Each Principal Component (SVD)

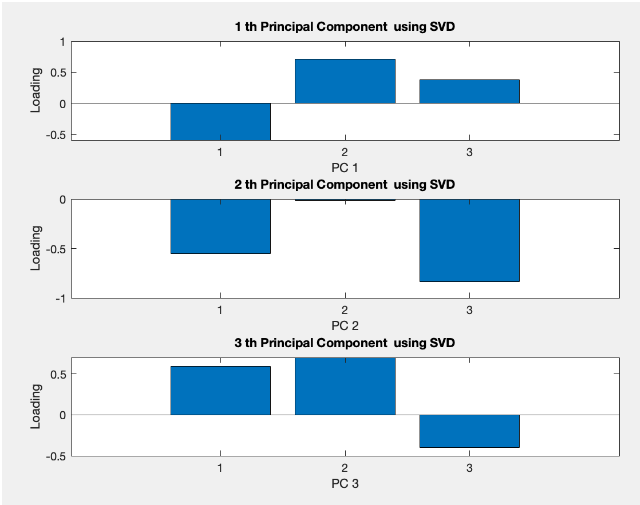


Figure 10: Principal Component Loadings (SVD)

3. **New Directions (Principal Components):** The new directions are derived from the left singular vectors (U) and plotted.

```
new_directions_svd =

   -0.5914   -0.5493    0.5903
    0.7121   -0.0124    0.7019
    0.3783   -0.8355   -0.3986
```

Figure 11: New Direction (SVD)

4. **Whitened Data:** The data is projected onto the principal components to get the whitened data and visualize it.

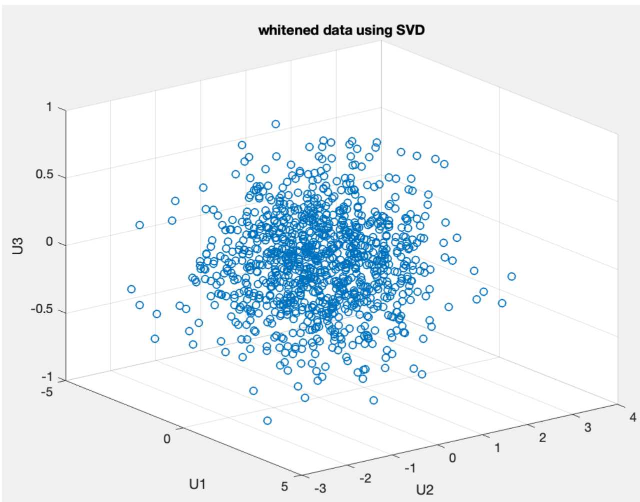
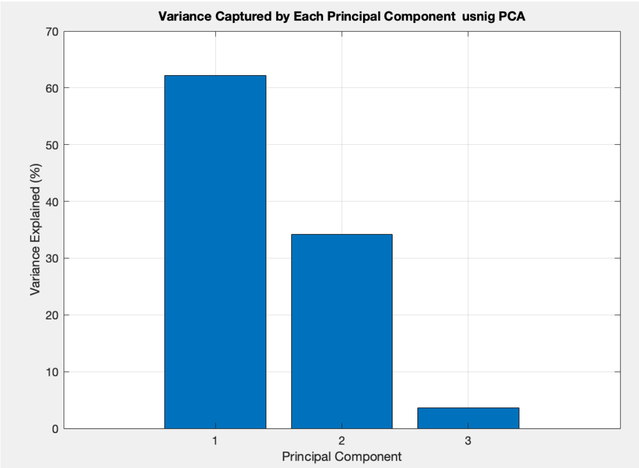


Figure 12: Whitened Data (SVD)

3.A.c Part (c): PCA Analysis Using MATLAB's PCA Function

1. **PCA Computation:** The PCA function in MATLAB is used on the data to get the principal component coefficients (coeff), scores, and explained variance.

2. **Explained Variance:** The variance explained by each principal component is calculated and plotted.




```
data_variance_explained_pca =

   62.1826
   34.1765
    3.6409
```

Figure 13: Variance Explained by Each Principal Component (PCA)

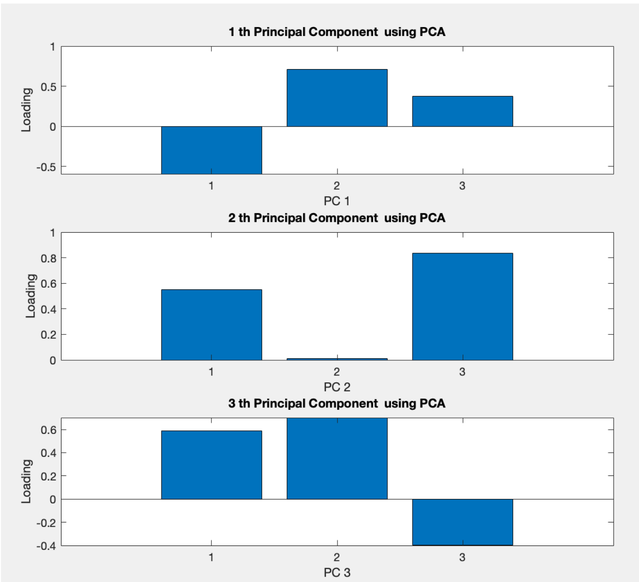


Figure 14: Principal Component Loadings (PCA)

3. **New Directions (Principal Components):** The new directions are derived from the principal component coefficients (coeff) and plotted.

```
new_directions_pca =

   -0.5914    0.5493    0.5903
    0.7121    0.0124    0.7019
    0.3783    0.8355   -0.3986
```

Figure 15: New Direction (PCA)

4. **Whitened Data:** The data is projected onto the principal components (coefficients of the pca function) to get the whitened data and visualize it.

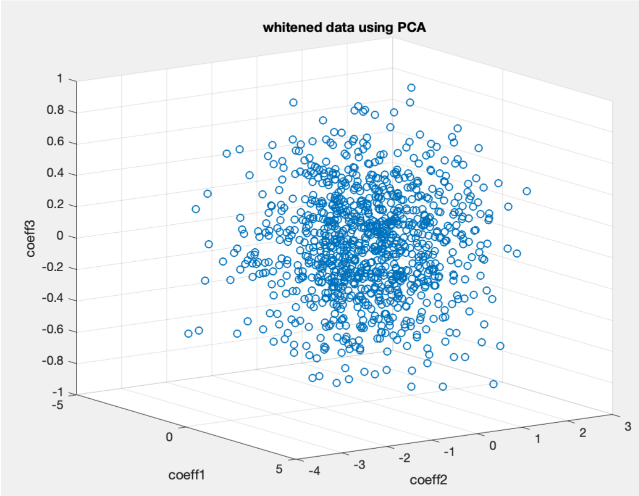


Figure 16: Whitened Data (PCA)

3.B Conclusion

Both SVD and PCA give similar results for new directions (principal components) and the amount of variance each principal component explains. Most of the variance is captured by the first principal component, with a sharp drop for the next components. The whitened data visualizations are similar for both methods, confirming the consistency of results from SVD and PCA.

4 Statistical Analysis Using SVD on House Price Dataset

In this question we want to analyze our `house_dataset.mat` dataset. The provided dataset contains 13 features for 506 houses, with the prices of these houses. The goal of this analysis is to extract meaningful statistical insights using Singular Value Decomposition (SVD). The dataset includes the following features for each house:

1. Per capita crime rate per town
2. Proportion of residential land zoned for lots over 25,000 sq. ft.
3. Proportion of non-retail business acres per town
4. 1 if tract bounds Charles River, 0 otherwise
5. Nitric oxides concentration (parts per 10 million)
6. Average number of rooms per dwelling
7. Proportion of owner-occupied units built prior to 1940
8. Weighted distances to five Boston employment centers
9. Index of accessibility to radial highways
10. Full-value property-tax rate per $10,000
11. Pupil-teacher ratio by town
12. $1000(Bk - 0.63)^2$, where Bk is the proportion of blacks by town
13. Percent lower status of the population

4.A Methodology

4.A.a Loading and Preprocessing the Data

The dataset is loaded and the features matrix (13 x 506) is extracted. The average price of houses is calculated to classify each house based on whether its price is above or below this average.

Houses are classified into two classes:

- Class 1 (209): Price $>$ Average
- Class 0 (297): Price $\leq$ Average

4.A.b SVD Computation

The data is normalized by subtracting the mean and dividing by the standard deviation. SVD is performed on the normalized data to obtain the matrices U, S, and V.

4.A.c Principal Component Analysis

The first five principal components are used to project the data for visualization. We project the data using every two combination of these five principal components.

The loadings for the first 13 principal components are shown in the following plots. These loadings indicate the contribution of each feature to the principal components. For instance, if a particular feature has a high loading in PC1, it means that feature heavily influences PC1.

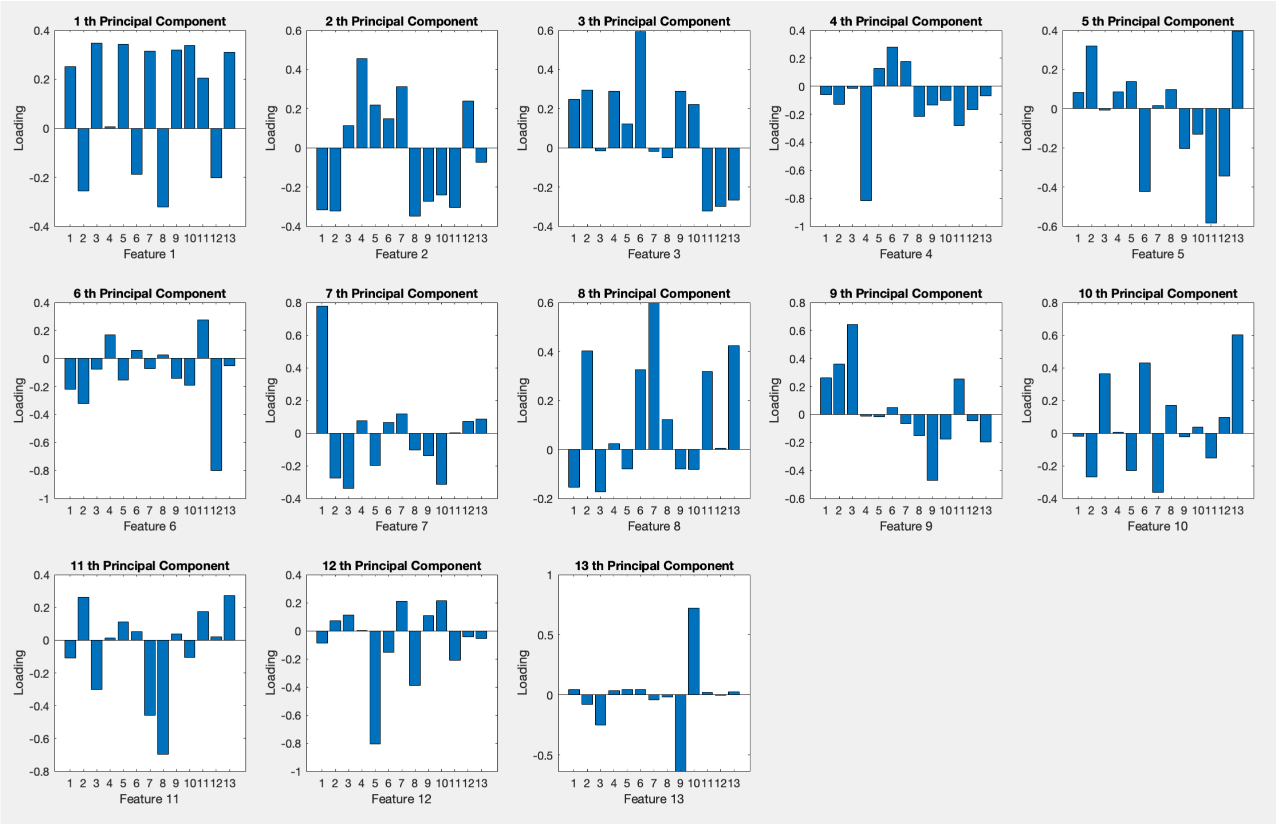


Figure 17: Principal Component Loadings

The explained variance by each principal component is calculated and plotted. The variance explained by each principal component is plotted below. The first principal component explains a significant portion of the variance, followed by a steep drop in the variance explained by subsequent components.

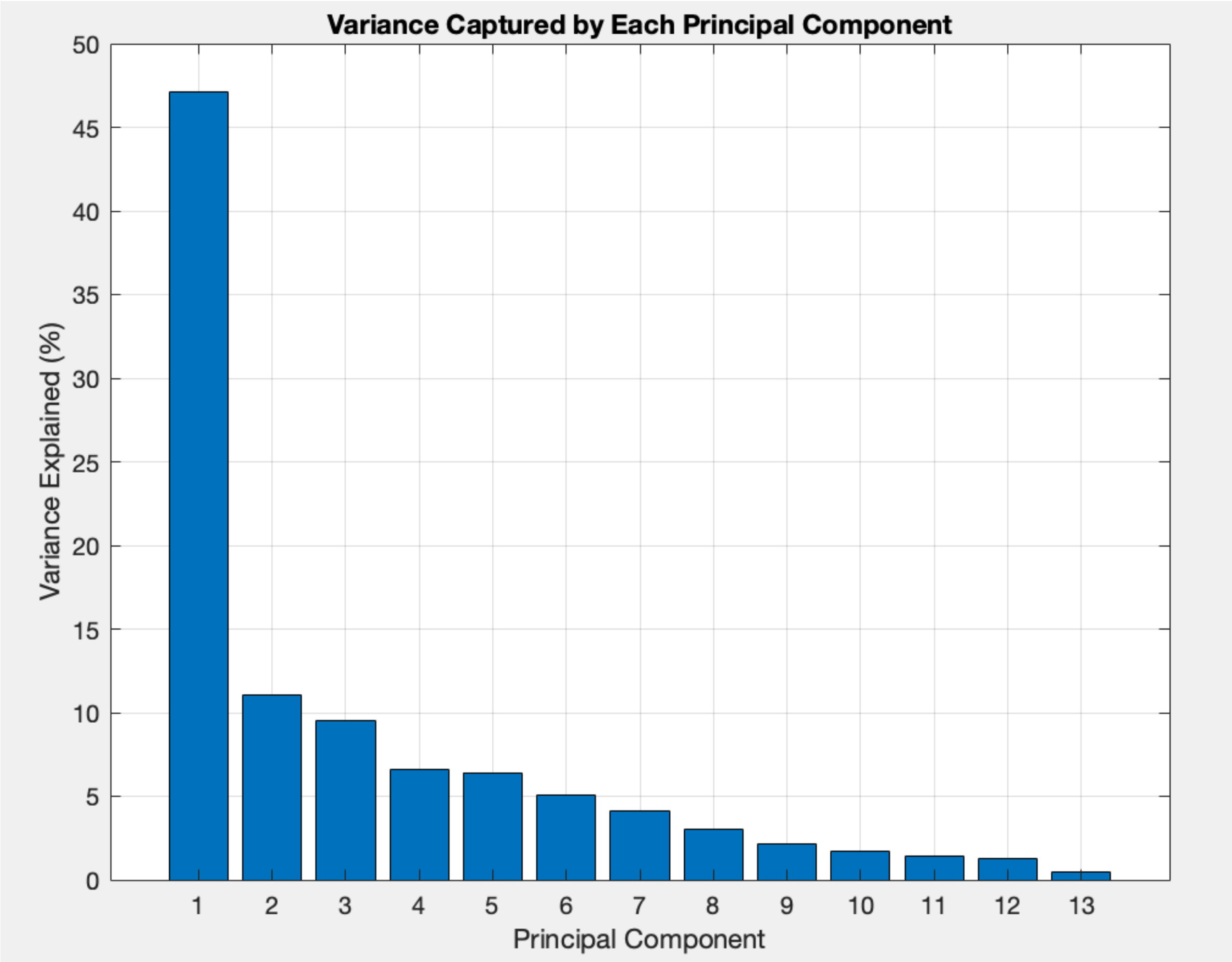


Figure 18: Variance Explained

Scatter plots for the first few pairs of principal components show the classification of houses based on their prices.

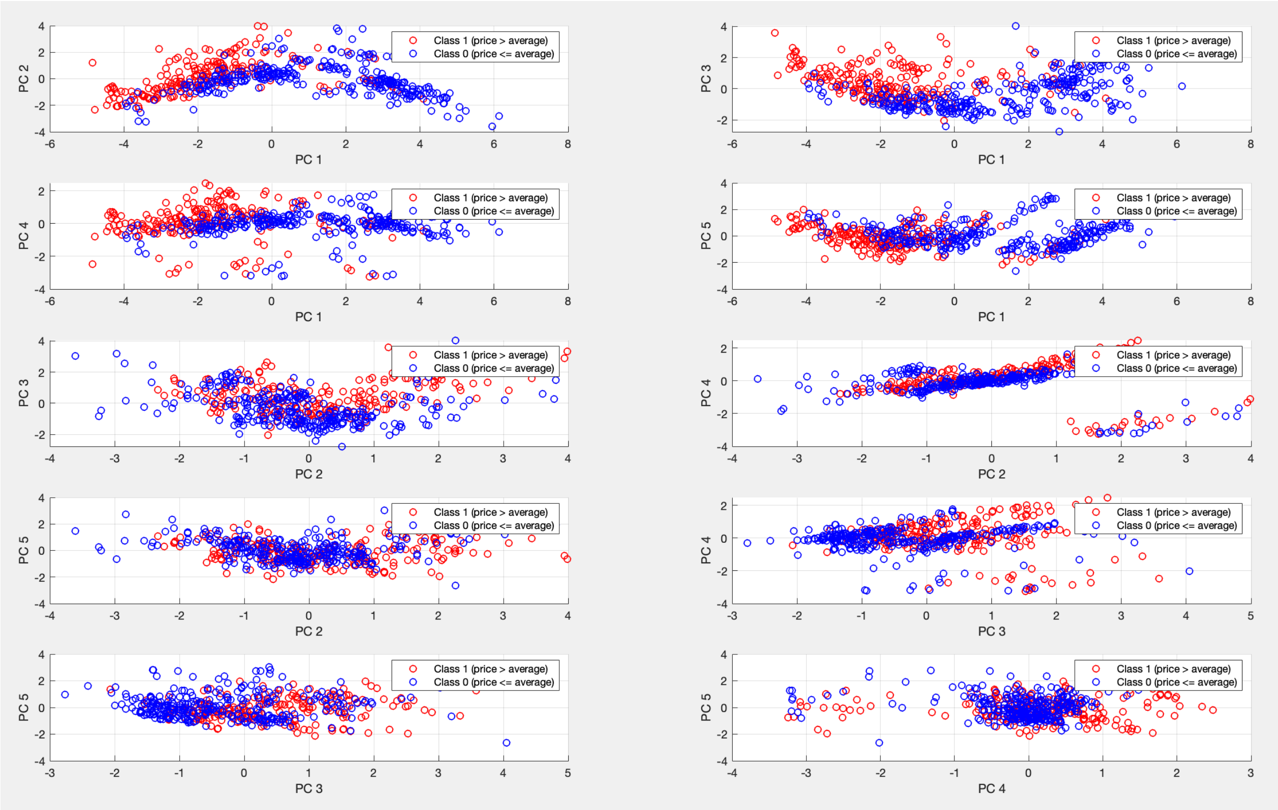


Figure 19: Scatter Plots

Class 1 (red) and Class 0 (blue) are somewhat separable along PC1, which is expected given the variance explained by PC1. The separation between Class 1 and Class 0 points is relatively clear along PC1. This indicated that the first principal component plays a crucial role in distinguishing expensive houses from cheaper ones. Plots involving other principal components (e.g., PC3, PC4, PC5) show more overlapping between the two classes, indicating that these components have less discriminative power compared to PC1.

4.B Conclusion

The SVD analysis of the house price dataset reveals that the first principal component captures the majority of the variance in the data. The scatter plots indicate a noticeable separation between houses with prices above and below the average when projected onto the first few principal components. The loadings plot highlights the contribution of each feature to these principal components, providing insights into which features are most influential in determining house prices.