University of Mannheim
Business Informatics and Mathematics
Fall 2023
IE 675b Machine Learning
Dr. Rainer Gemulla

Assignment 3: Singular Value Decomposition

November 26, 2023

Laila Albalkhi
Jonah Niklas Bjørgsvik Wiecek

lalbalkh | 1968154 | albalkhl@uwindsor.ca
jwiecek | 1966868 | jwiecek@students.uni-mannheim.com

Contents

1 Intution on SVD

1.1 Looking at the Data

$$M_1 = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{1}$$

We can observe the rank of M_1 to be 1, the number of linearly independent rows. By looking at the matrix, we can construct two vectors U_1 and V_1 that describe the data, where V_1 can be interpreted as the representative part or prototype, and U_1 describes the composition of the data in terms of parts. Since the vectors need to be of unit norm, we can then normalize these vectors by dividing them by their norm. We see this here:

$$U_1 = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \end{pmatrix} \qquad V_1^T = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \end{pmatrix} \tag{2}$$

$$\hat{U_1} = \begin{pmatrix} \frac{1}{\sqrt{3}} \\ \frac{1}{\sqrt{3}} \\ \frac{1}{\sqrt{3}} \\ 0 \\ 0 \end{pmatrix} \qquad \hat{V_1^T} = \begin{pmatrix} \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} & 0 & 0 \end{pmatrix} \tag{3}$$

Given that the matrix is of rank 1, we know that there will only be 1 singular value. Taking a look at the first non-zero value in M_1, we know that the singular value σ must fulfill the following equation:

$$(\frac{1}{\sqrt{3}})(\sigma)(\frac{1}{\sqrt{3}}) = 1 \tag{4}$$

$$\text{Thus, } \sigma = 3 \tag{5}$$

We can follow very similar steps for matrices M_2 and M_3.

$$M_2 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 1 & 2 & 0 \\ 0 & 2 & 1 & 2 & 0 \\ 0 & 2 & 1 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{6}$$

The compact single value decomposition for M_2 becomes the following:

$$\hat{U_2} = \begin{pmatrix} 0 \\ \frac{1}{\sqrt{3}} \\ \frac{1}{\sqrt{3}} \\ \frac{1}{\sqrt{3}} \\ 0 \end{pmatrix} \qquad \hat{V_2^T} = \begin{pmatrix} 0 & \frac{2}{3} & \frac{1}{3} & \frac{2}{3} & 0 \end{pmatrix} \qquad \sigma = 3\sqrt{3} \tag{7}$$

$$M_3 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{pmatrix} \tag{8}$$

The compact single value decomposition for M_3 becomes the following:

$$\hat{U_3} = \begin{pmatrix} 0 \\ \frac{1}{2} \\ \frac{1}{2} \\ \frac{1}{2} \\ \frac{1}{2} \end{pmatrix} \qquad \hat{V_3^T} = \begin{pmatrix} 0 & \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} \end{pmatrix} \qquad \sigma = 2\sqrt{3} \tag{9}$$

$$M_4 = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix} \tag{10}$$

With M_4, we now have a rank of 2, since there are 2 linearly independent rows. We construct U

and V as follows and normalize them respectively. This also means that we have two singular values, which we can find using the original data matrix.

$$\hat{U_4} = \begin{pmatrix} \frac{1}{\sqrt{3}} & 0 \\ \frac{1}{\sqrt{3}} & 0 \\ \frac{1}{\sqrt{3}} & 0 \\ 0 & \frac{1}{\sqrt{2}} \\ 0 & \frac{1}{\sqrt{2}} \end{pmatrix} \qquad \hat{V_4^T} = \begin{pmatrix} \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{pmatrix} \qquad \sigma_1 = 3 \qquad \sigma_2 = 2 \tag{11}$$

$$M_5 = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \end{pmatrix} \tag{12}$$

For M_5, we can observe a rank of 3 and the respective 3 linearly independent rows, as follows:

$$U_5 = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix} \qquad V_5 = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \end{pmatrix} \tag{13}$$

While we can normalize them, they are not orthogonal (namely V_5 in this case). This means we can't use the same strategy as before to find representative rows and columns of the data. We defer to NumPy calculations in this case and give up the manual calculations.

$$M_6 = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix} \tag{14}$$

We observe a rank of 2 for M_5 but again run into the orthogonality requirement problem. Given the rank, we assume there will be 2 singular non-zero values. We defer to NumPy calculations in this case and give up the manual calculations.

1.2 NumPy Comparison

The first 4 matrices were correct when using NumPy to compare the SVD computation in terms of the non-zero singular values and their corresponding left and right singular vectors.

1.3 Rank-1 Approximation

The best rank-1 approximation is intuitive and accurate for most of the matrices provided. The best rank-1 approximation is the matrix itself if the rank is 1, and when the rank is 2 we can see that it corresponds to the first principal component of the matrix.

We take a closer look at M_5 and M_6. The matrix that we see is corresponding to the first principal component. For M_5, the component has more mass assigned to the center and less for the sides. This is because every entry of U has a 1 in the center position. But 4 of the rows have 0's either to the right or left of the center. The third data point assigns a higher weight to this part since it has 1's across all features. It is the opposite for M_6, where all of the data points/rows have 1's in for all features except for the center one. Hence the first principal component has the same form as in M_5 just inverted; less mass in the center, more mass off-center.

1.4 Closer Look at M_6

As previously discussed, we can observe M_6 to have rank 2, which means it should also have 2 non-zero singular values. However, the NumPy SVD computation returns 5 non-zero singular values. By looking further into the $svdcomp$ function, we realize that it is due to floating-point arithmetic. This is confirmed by the first two singular values being the largest, whereas the rest are extremely close to 0.

2 The SVD on Weather Data

2.1 Normalization

Normalization is reasonable considering our data because we have features that are different both in scale and range. Our data consists of temperature and rainfall data, where temperature ranges from -30 to 30 Celcius and rainfall ranges from 0mm to 310mm. Normalizing will allow us to use both metrics within the same data set.

Normalization is especially useful if we want to use clustering with K-means since it is a distance-based algorithm. Additionally, centering the data draws an equivalency between SVD and PCA later

on. When we plot the histogram we see the exact same distributions as before just with a change to the x-axis. This is as expected since we are not transforming the distribution, only rescaling it.

2.2 Computing the SVD

We compute the SVD with NumPy and find that it is full rank. This is to be expected from a dataset with real-world samples. In later sections, we will see that many of the singular values are quite small; we can truncate the SVD without losing too much of the original data.

2.3 First 5 Columns of U

The latitude and longitude were plotted with the colouring of the left singular vectors. This can be interpreted as how many of the right singular vectors are used in each data point. This was done for the first 5 singular vectors.

In the first plot, we observe that the northern points tend to have positive weights, the southern points tend to have negative weights, and the border is a grey area with weights closer to zero. When we study the corresponding right singular vector (first row of Figure 2.1), we see that all of the temperature data is assigned negative weights of roughly -0.2 and the rain data is assigned weights closer to zero with the exception of the rain data for June-September, which have a positive weight of roughly 0.1.

This shows us that the first singular vector mainly resolves around distinguishing warmer and colder regions, as well as regions that experience more rainfall during the summer. The North-South split we see on the map[1] reflects this as well.

The second singular vector seems to distinguish East to West. And it also highlights many areas that are along the coast. Therefore we were thinking this could be associated with rainfall. This is confirmed by studying the second row of V^T.

The third seems to focus on rain during the summer, the fourth and fifth components are harder to interpret. This also makes sense since they have decreasing importance or weights (singular values) corresponding to them. So we interpret the other principal components as being "corrections" that can be used against the errors that are made by the most important components.

[1] The plot of latitude and longitude coloured with the values of the left singular vectors

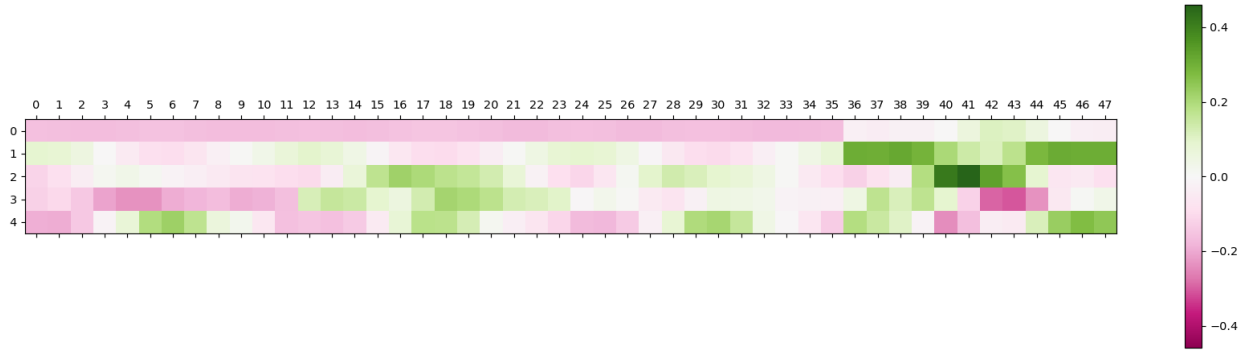

Figure 2.1: The five first right singular vectors

2.4 Scatterplots

By plotting the data on the first two left singular vectors instead of coordinates, and then colouring with either centred latitude or longitude, we can see that they are fairly well-separated. The data points from the north tend to have a high value in the first singular vector and the points from the south a low. The scatterplot colored with longitude is not as well separated, but there still is a clear correlation with the second left singular vector. The Eastern points tend to have negative weights while the data points from central Europe are a bit mixed, and the Western points have positive weights.

When we take a look at some other left singular vectors plotted, we see that the North-South and East-West distinction is not as present as in the first two left singular vectors. From this, we can derive that the most important aspect of differentiating between different climates is actually the geographical location.

2.5 Truncated SVD

For this task mainly see implementation.

2.5.1 Guttman-Kaiser

There were 37 singular values greater than 1. Guttman-Kaiser suggests k=37.

2.5.2 Frobenius Norm

The first three singular values have 93% of the squared Frobenius norm of the full Σ. Hence $k = 3$ is another suggestion.

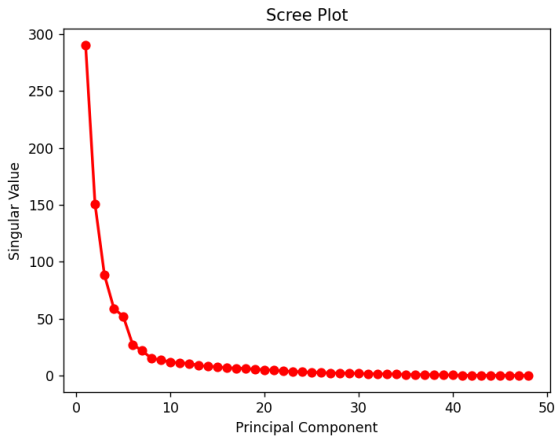

Figure 2.2: Scree plot

2.5.3 Scree Test

For the Scree test, we are simply plotting the singular values. Here we can see that there are huge drops in the beginning, but at k=6 we are already past the "Scree". Since this is a subjective method it is up to us to choose k. $K = 6$ could be a choice, but we could also argue that we see a large difference from 1 to 3 components but not as big from 3 to 6. Hence, the plot strengthens the suggestion from the Frobenius norm method.

2.5.4 Entropy

In the entropy method we pick k such that $\sum_{i=1}^{k} f_k > E$, where $f_k = \frac{\sigma_k^2}{\sum_i \sigma_i^"}$. From this method, we also get $k = 3$.

2.5.5 Random Sign Flipping

In the random flipping of signs, we are using the residual matrix after size-k truncation. Since the Frobenius norm is not affected by flipping the signs, we use the spectral norm.
We see a minimum in the plot when we use the $k = 7$ first principal components. This is another potential choice.

2.6 Root-Mean-Square Error (RMSE)

In the exercises, we showed that the squared Frobenius norm of the error is the same as the sum of the squared singular values that we don't include in our k-truncation.
Therefore we can simplify the calculation of RMSE to the following when we are using an approximation of A that is a k-truncation of the SVD:

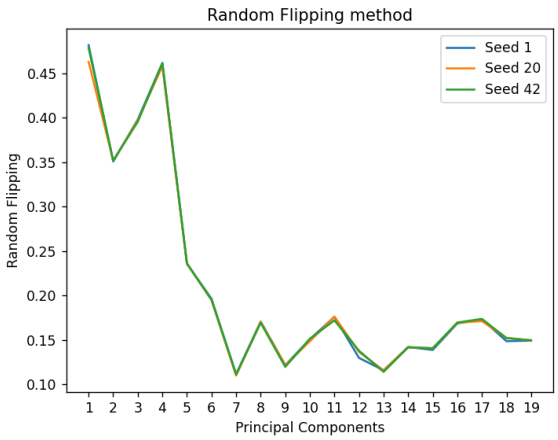

Figure 2.3: Result for a random flipping method with k up to 20

$$rmse(A, \tilde{A}) = rmse(A, \boldsymbol{\sigma}, k) \tag{15}$$

$$= \frac{\sqrt{\sum_{i=k+1}^{min(m,n)} \sigma_i^2}}{\sqrt{mn}} \tag{16}$$

We choose 10 values for epsilon in the range $[0, 2]$ and compute the SVD after the noise has been added to X. See code for implementation.

In Figure 2.4 we clearly see that if we use k as the full rank we of course get no error since $A = A_k$. Otherwise the RMSE is inverse proportional to k. The fewer principal components we use the more error we get. We also see that the noisier the data gets the more principal components are needed to explain the data. This is also intuitive because when we add noise, the inherent structure of the data gets reduced.

3 SVD and Clustering

3.1 Clustering Explanation

Given that the data contains temperature and rainfall information, we can interpret the clusters as different climates. Each cluster represents latitudinal and longitudinal points that experience the same temperature and rainfall. For example, the pink cluster could represent climates where there is low rainfall and high temperatures in the summer season, whereas the light green cluster could represent large amounts of rainfall and low temperatures in the winter.

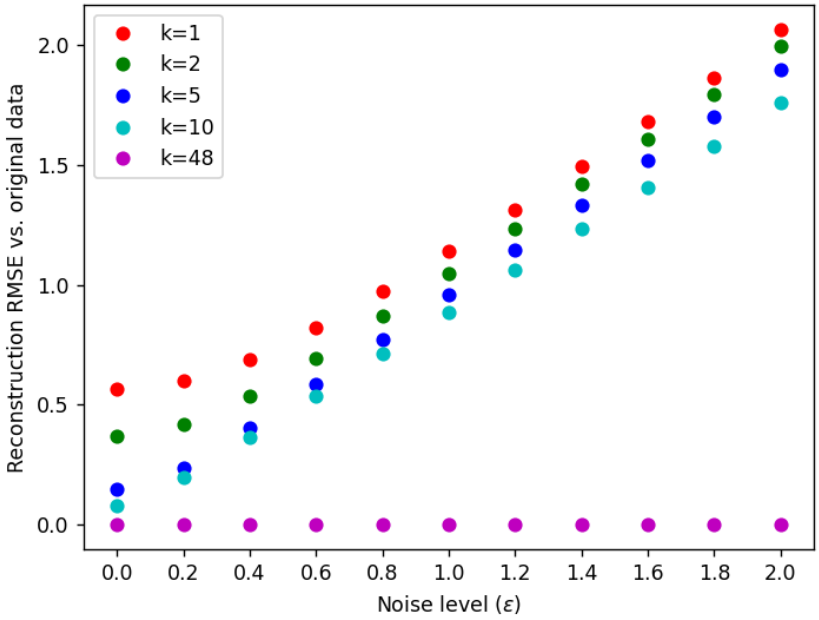

Figure 2.4: Error of reconstructed Matrix X with varying noise

3.2 Left vs. Right Singular Vector

As we can see in Figure 3.2, when we plot the first and second left singular vectors, we see well-separated clusters that move along the x-axis. We see that given just the first left singular vector, we can indeed identify 4 out of 5 of the clusters since they are divided horizontally. The light green cluster/climate however is not distinguishable purely by the first left singular vector, but it is almost separable by looking at the points with higher than 0.2 in the second left singular vector. If we take both into consideration, we can find pretty clear boundaries between all of them. This can be interpreted as having knowledge about the two first principal components (the general temperature rainfall of a region) is enough to determine its climate.

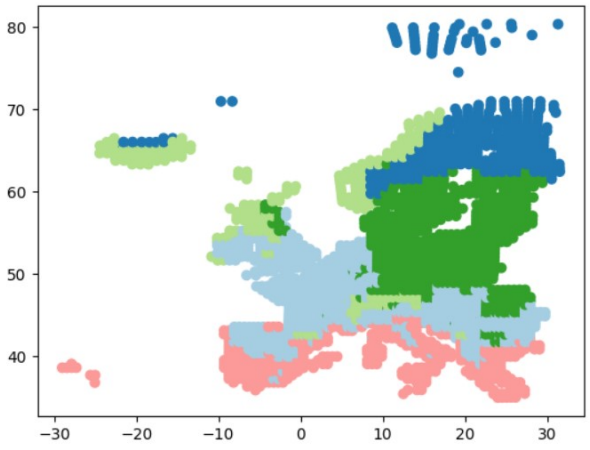

Figure 3.1: Clustered data by latitude and longitude.

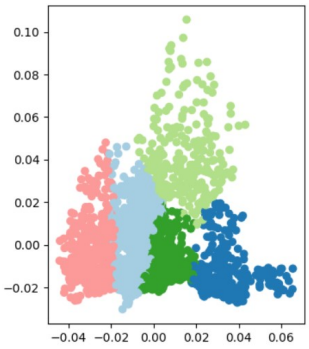

Figure 3.2: Clusters by the first two left singular vectors.

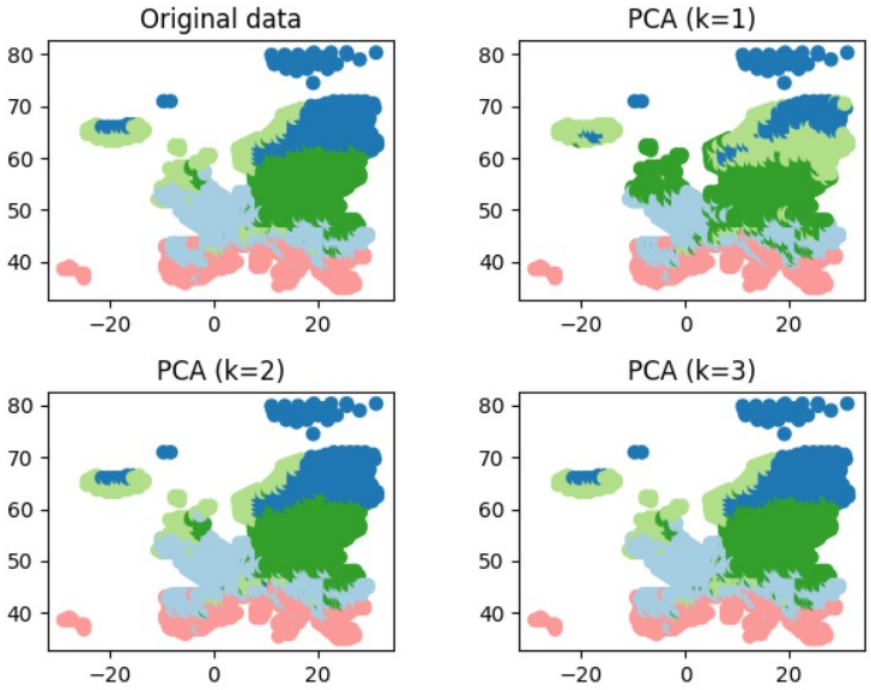

Figure 3.3: Caption

3.3 PCA Scores

We compute the PCA scores and reduce the data to the first 3 components, comparing with the original data. When $k = 1$ in Figure 3.3, we can see that the clusterings change. This is because we are only using the first left singular vector, the first singular value, and the first right singular vector to represent the entirety of the dataset. However, this extreme dimensionality reduction results in a loss of information.

However, when $k = 2$ or $k = 3$, we see less of a difference in the original clustering. If we refer back to the Forbenius norm calculation we did earlier, we realize that this is because there is already 86% of the data represented when we truncate to $k = 2$, with smaller increments. We can take a look at the scree plot as well to verify and confirm this. In conclusion, the last two PCA scores do well to approximate the entirety of the data even with dimensionality reduction.