Assignment 3: SVD(Singular Vector Decomposition)

Seoyoung Yoo
Matrikel-Nr. 1904668
20 November 2022

1 Intuition on SVD

a) Try to manually obtain the rank of each of the following matrices, as well as its singular values, and the left and right singular vectors corresponding to the non-zero singular values. Do this by "looking" at the data and try to infer how the compact SVD needs to look like.

How to calculate SVD Method:

$$A = U\sum V^T$$

$$A^TA = (U\sum V^T)^T U\sum V^T = V\sum^T\sum V = V\sum' V$$

$$A^TAV = V\sum' V^TV$$

$$(A^TA)V = V\sum'$$

$$AA^T = U\sum V^T\ (U\sum V^T)^T = U\sum V^TV\sum^T U^T = U\sum\sum^T U^T = U\Sigma'' U^T$$

$$AA^TU = U\Sigma'' U^TU$$

$$(AA^T)U = U\Sigma''$$

b) Compute the SVD (e.g., using NumPy) and compare. Were you correct?

Using above method of calculating, I get different result as using NumPy.

c) What does the best rank-1 approximation look like? Is it "intuitive"?

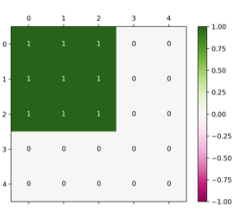

Figure 1. M1

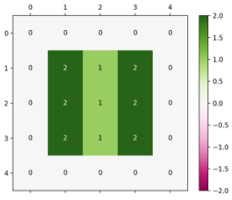

Figure 2. M2

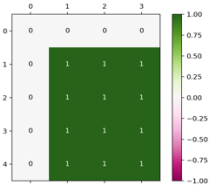

Figure 3. M3

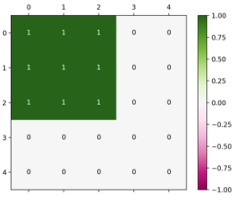

Figure 4. M4

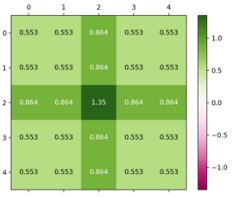

Figure 5. M5

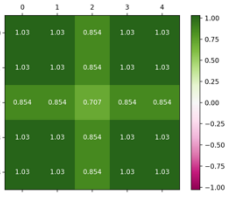

Figure 6. M6

If we look at Figures 1 to 3, you can see that the overall intuitive approximation. However, looking at Figures 4 through 6, it can be seen that the actual values of M4, M5 and M6 are approximately differen t. This can be inferred that the NumPy library calculation process is different. In the usual convention

the singular values are sorted in descending order so that the first singular value is the large st one and thus the first term of the SVD the best rank-1 approximation of the matrix.

d) How many non-zero singular values does M6 have, i.e., what is the rank of M6? How many non-zero singular values are reported by NumPy? Discuss!

If we check the singular value of M6 using the np.diag() method on the NumPy library, you can see that it is 5, but if you check it using the np.linalg.matrix_rank() method, it is output as 2. It was confirmed through documentation [1]that the cause of this error was due to handling floating-point rounding errors in the SVD calculation when calculating the corresponding method of the NumPy library.

2 The SVD on Weather Data

a) Normalize the data (climate matrix) to z-scores and store the result in variable X. Considering the data we are using, are the assumptions for normalizing the data reasonable?

How to calculate z-scores:

$$z - scores = \frac{X - \mu}{\sigma}$$

SVD is sensitive to data scaling. Since the scales of the features of 'data/worldclim.csv' are different, it is reasonable because the goal of normalization is to ensure that all data points are reflected at the same scale (importance). Normalization using z-scores avoids outlier problems.

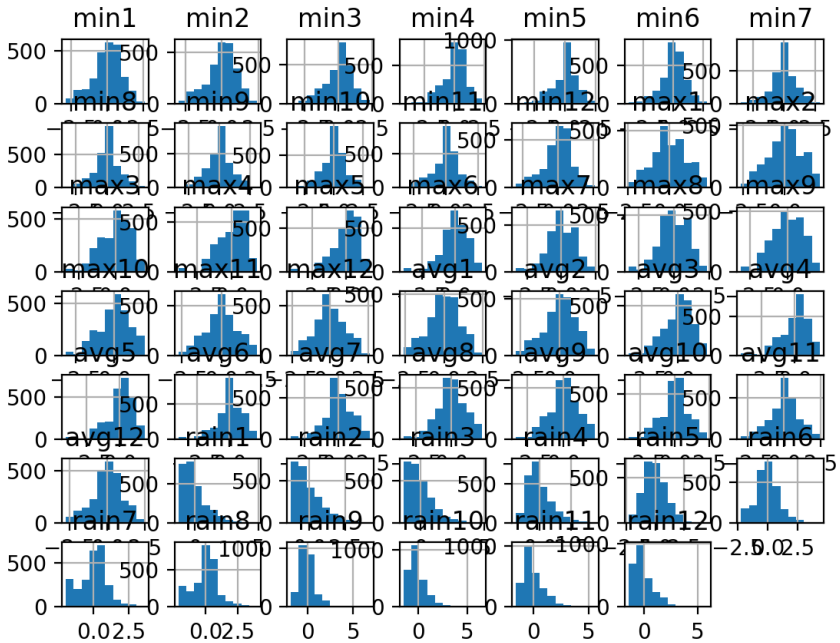

Figure 7

[1] NumPy documentation Link: https://numpy.org/doc/stable/reference/generated/numpy.linalg.matrix_rank.html

b) Compute the SVD $U \Sigma V^T$ and the rank of the normalized data.

The rank of normalized climate data is 48.

c) Plot each of the first 5 columns of U. Use the longitude and latitude of each data point as the x and y coordinates, respectively, and the corresponding entry in the left singular vector to color each point (see provided code). Can you interpret the result?

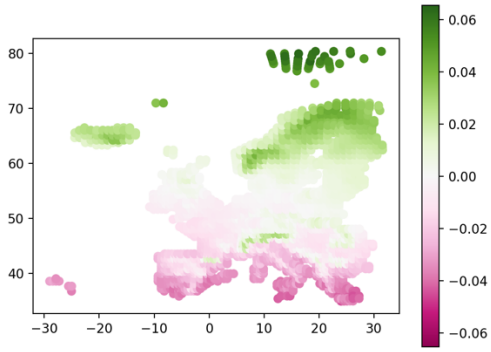

Figure 8

d) Plot some scatterplots between the columns of U using colors to distinguish either their North South or East–West location (see provided code). Can you interpret the results?

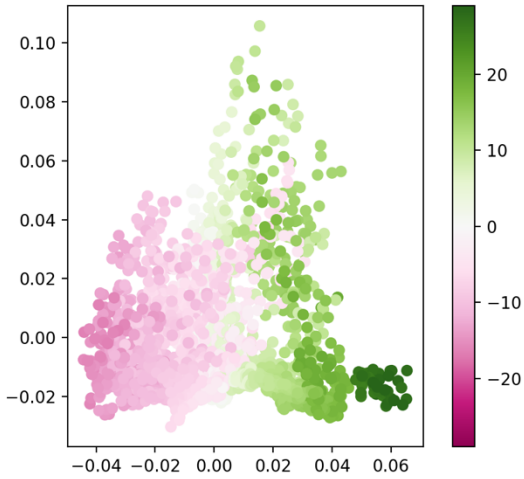

Figure 9

e) Try the different rank selection methods listed below to decide what would be a good size for a truncated SVD. Report the size each method suggests (and when subjective evaluation is needed, say why you picked your choice).

(i) Guttman–Kaiser criterion: 37
(ii) 90% of squared Frobenius norm: 3
(iii) Scree test: 4

(iv) Entropy-based method: 1
(v) Random flipping of signs: 7

What, if any, would be your ultimate choice? Scree test looks the most reasonable.

(i) **The Guttman-Kaiser** method determines how many eigenvalues of the sample correlation coefficient matrix are greater than 1 because the minimum dimension (rank) of the matrix coincides with the number of eigenvalues greater than 1. In a small sample, the number of factors tends to overestimate and under/overestimate the number of factors depending on conditions. Although it can be used as an approximation to determine the number of factors, it is inappropriate to use as a criterion for determining the final number of factors.

(ii) **90% of squared Frobenius norm*** is to select enough singular values such that the sum of their squares is 90% of the total sum of the squared singular values.

*The Frobenius norm is a norm often used when you need to find the size of a matrix.

$$\| A \|_F = \sqrt{\sum_{i,j} A_{i,j}^2}$$

(iii) **Scree test** is one of the methods for determining the number of factors to be extracted from the correlation matrix being analyzed when performing factor analysis. Since each eigenvalue reflects how much each dimension explains the data, adding a new dimension does not significantly increase the explanatory variance, so the factor is not included. After sorting the eigenvalues in descending order, the number of factors should be determined at the point where the eigenvalues do not increase when a new dimension is added. (Figure 10)

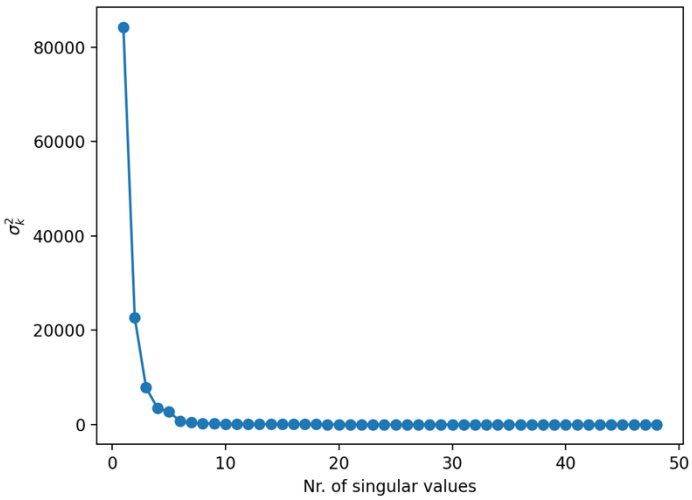

Figure 10

(iv) **Entropy-based method** is that entropy indicates the degree of disorder, and when there are few types of variable values, entropy is low.

(v) **Random flipping of signs** is to select *k* such that the residual matrix contains only noise. (Figure 11)

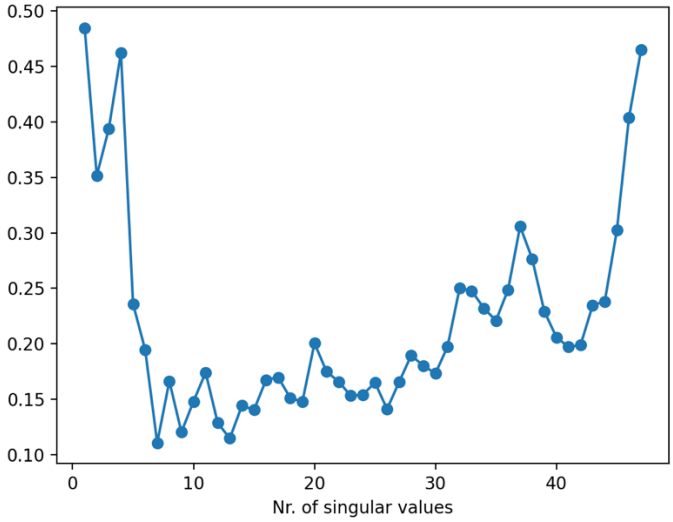

Figure 11

f) **Define the root-mean-square error (RMSE) between an m × n matrix A and an m × n approximation $\hat{A}$**

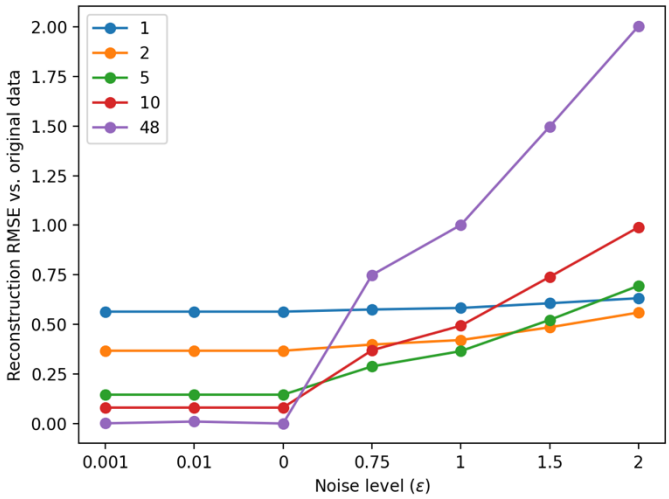

Figure 12

3 SVD and Clustering

a) Look at the resulting clustering and explain what the clusters may represent (remember, the data contains temperature and rainfall information).

If we look at Figure 13 below, you can see that 5 clustering was done well according to the temperat ure info according to the latitude.

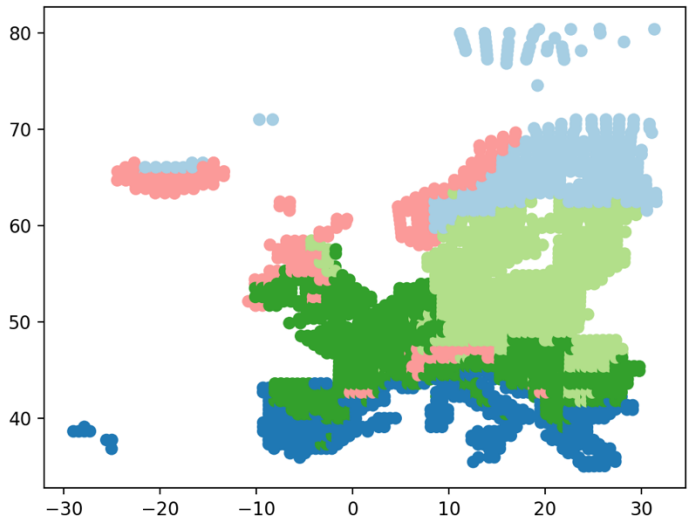

Figure 13

b) For another visualization of the results, plot the data so that the x-axis position comes from the first left singular vector, the y-axis position comes from the second left singular vector, and the color of each point is defined by the cluster identifier. Are the clusters well-separated from each other in the plot or are they mixed? Do some of the clusters look like outliers?

As can be seen in Figure 14 below, the x-axis position comes from the first left singular vector, the y-axis position comes from the second left singular vector, and as a result of clustering, it can be observed that it is well-separated.

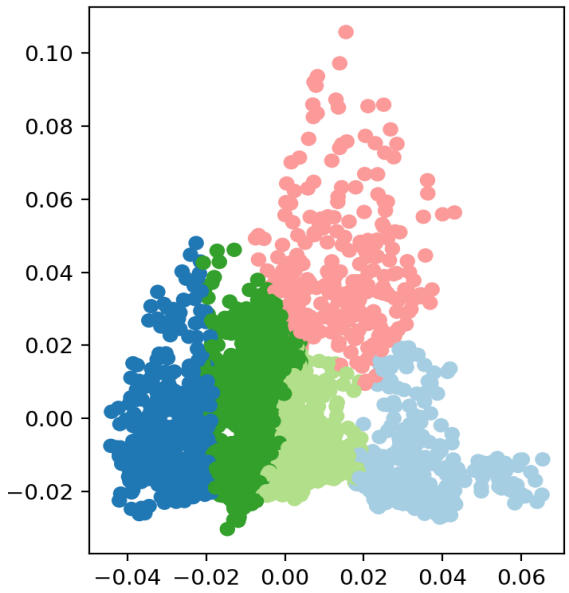

Figure 13

c) Compute the PCA scores of the data points (in X) for the first k principal components for k ∈ {1,2,3}, thereby reducing dimensionality to k. Do this solely (!) using the SVD of the appropriate version of the

data. Repeat the clustering and visualization steps of a) with this new data. Did the results change? Why do you think the results changed or did not change?

As can be seen in Figure 15 below, it can be seen that the original data and the clustering result of PCA are the same.

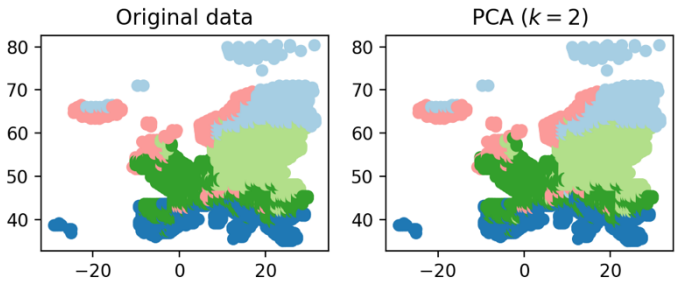

Figure 15