Matlab session 5: Singular value decomposition and principal component analysis

Alexander Bertrand

December 9, 2019

1 Image compression

In this exercise, we will use the singular value decomposition for the purpose of compression. We provide the script "Ex1.m" which is partially completed.

1. Open "Ex1.m", and run the first part to visualize an image.

2. Compute the SVD of the image to obtain U, Σ and V.

3. Reorder the singular values in decreasing order and reorder the matrices U and V accordingly. Plot the singular values.

4. Reconstruct the matrix keeping only the 5, 10, 15, 50, 70 and 100 largest singular values. For this, complete the code within the for loop in the second part of the code.

5. Compute the compression ratio for each different case (complete the code within the loop). The compression ratio is the ratio between the number of entries to store after compression and before.

6. How many singular values would you choose for a good reconstruction (explain both visually and using the singular value plot)? Explain how this is a compression.

7. You can try this exercise with other images of your choice.

2 Singular value decomposition

Create an m-file that executes the following tasks:

1. Create a random 20×2 matrix B and a random 2×100 matrix C using the command `randn`. Then compute the matrix A as $A = BC$. What will be the rank of A? (first answer yourself and then check with Matlab).

2. Create a 20×100 noise matrix N as `N=randn(20,100)` and create a matrix `A_noisy=A+N`. What will be the rank of `A_noisy`? (first answer yourself and then check with Matlab).

3. Compute the SVD of `A` and the SVD of `A_noisy` using the command `svd`. Take two left singular vectors and check whether they are orthonormal. Do the same for two right singular vectors.

4. Create a matrix Q for which the columns contain an orthonormal basis of the null space of A (extract Q from the SVD). How many columns should Q have? Make a random linear combination of the columns of Q, store them in a vector $\mathbf{x}$, and verify that $A\mathbf{x} = 0$.

5. Create a figure, in which you plot the singular values of `A` (in blue) and the singular values of `A_noisy` (in red), both ordered in descending order. Explain what you see.

6. Note that in the red plot, two singular values are much larger than the others. This indicates that there is an 'almost' rank-2 structure in `A_noisy`. Create a new matrix `A_approx` which is the *best rank-2 approximation* of `A_noisy`.

7. Compute the squared error between the entries of `A_approx` and `A_noisy`, i.e., $\sum_{i,j}(a_{ij} - b_{ij})^2$ where a_{ij} and b_{ij} are corresponding entries in the two matrices.

8. Compute the sum of the *squared* singular values of `A_noisy` that were removed to obtain the rank-2 approximation. Compare this sum with the squared error from the previous step. What do you observe?

9. Create a figure, in which the first row of `A` is plotted (in blue), and where the first row of `A_noisy` (in red) and the first row of `A_approx` (in green) are plotted on top .

10. Can you observe the denoising properties of the SVD in the last figure? Show that this works for all rows by comparing the squared error between (`A_noisy` and `A`) with the squared error between (`A_approx` and `A`).

11. Would such a denoising still work if the original random data matrices B and C were both rank 20?

3 Variance compactification by Principal Component Analysis

Create an m-file that executes the following tasks:

1. Load the file `cloud.mat`. This file contains a matrix X with $N = 500$ datapoints in R^3.

2. Make a scatter plot of the 500 data points, represented as black dots in 3D space (Hint: use the command `plot3`, and check its help file to understand how to use it. It may also help to check the 'LineSpec' option in the help file of the command `plot` to see how you can plot individual points without connecting lines.).

3. On the same plot, indicate the mean of the data with an asterisk (*) in red.

4. Make a new matrix B, which is in mean-deviation form (this means that you subtract the mean from each data point).

5. Create a new figure, with the same scatter plot as before, but now using the data points in the mean-deviation form. This means that the cloud of data should be centered around the origin.

6. Compute the 3 principal components of the data.

7. Perform a change of variables by projecting the 3D data on the three principal components, yielding three new variables `y1`, `y2`, `y3`.

8. Compute the variance of `y1`, `y2`, and `y3` (use the command `var`).

9. Compute the three eigenvalues of the sample covariance matrix S defined in Section 7.5 in Lay ($S = 1/(N-1)BB^T$), and compare with the values you obtained in the previous step. Explain what you observe.

   Hint: note that the principal components define the matrix P in an orthogonal diagonalization of S. Consider the variable $\mathbf{y} = P^T\mathbf{x}$ and its sample covariance matrix $1/(N-1)(P^TB)(B^TP) = P^TSP$. Use the fact that the diagonal elements of a (sample) covariance matrix contain the variances of the individual entries of a multi-variate stochastic variable.

10. Plot three lines through the origin in the direction of the three principal components (both the positive and negative parts). The length of each line should be equal to (2 times) the variance of the data in that particular direction. Use red for the first, blue for the second, and green for the third (you can again use the command `plot3` here, but connect each point with a line).

11. Make sure all axes of your plot cover an equal range (e.g., from -4 to 4). Hint: use the command `axis`.

12. After plotting, add the following commands:

    - `axis square` (makes sure that the axes have the same scale)
    - `grid on` (adds grid lines)

13. Check by visual inspection: The three lines should be orthogonal (why?). The red line should point in the direction with highest variance (why?). The green line should point in the direction with least variance (why?).

4 Spectroscopy revisited

In the previous Matlab session on least squares, we looked at the application of spectroscopy (e.g., to identify the relative amount of particles of a specific chemical component or biological tissue in a sample). In that session, we knew the clean spectral shape of two components as prior knowledge. In this session, we assume that this a-priori knowledge is not known, i.e., we only have access to the noisy data.

Create an m-file that executes the following tasks:

1. Load the file `spectraldata_noisy.mat`. This file contains a matrix X with $N = 2000$ spectra of biological tissue samples in its columns, and we know from the experimental set-up that each spectrum contains a mixture of two bio-chemical components. Note that each spectrum consists of 512 entries, i.e., we are analyzing data points in R^{512}. Therefore, we cannot make a visual scatter plot of the data (as in the previous exercise).

2. Plot the first spectrum. Run the m-file at this point, and observe that the spectrum is very noisy. In fact, it is too noisy to accurately quantify the relative amount of each of the two components in the sample.

3. Now let's try to find some structure in the data using PCA. Apply PCA to the data[1] in X, and generate a plot from which you can infer the number of principal components that are needed to explain most of the variance in the data (which plot should you make?). Run the m-file at this point, to find this number (you should 'see' that the answer is 2).

4. Put the first two principal components in the columns of the matrix $P = [\mathbf{u}_1\ \mathbf{u}_2]$. Plot both principal components on top of each other in two different colors. These are the two basis vectors on which we will project our data.

5. Compute the matrix $Y = P^T X$. Note that this projects the data onto the two first principal components to reduce the data points in R^{512} to points in R^2.

6. Compute the matrix $\tilde{B} = PY$, i.e., transform the data points in R^2 again to points in R^{512} by using them as coefficients with respect to the (orthonormal) basis formed by the two first principal component.

7. Reconstruct the data points in X from the matrix $\tilde{B}$ (by adding the mean again), and put the result in the matrix $\tilde{X}$.

8. Create a new figure. Plot the first spectrum in X (in blue) and the first spectrum in $\tilde{X}$ (in red) on top of each other. You should now be able to distinguish two resonance peaks in the spectrum. These can be used to identify the ratio with which the two main components are mixed in the tissue sample. Note how the noise is clearly reduced.

9. We have hidden an outlier sample somewhere in the data corresponding to a different tissue sample than the rest. Use PCA to find it, and plot the (original) spectrum of the outlier. Can you see why it is an outlier?

10. Create a new figure. Plot all *denoised* spectra in $\tilde{X}$ on top of each other in a single figure. Can you recognize the common structure they have?

[1] Don't forget to subtract the mean!

(they should all show peaks in the same resonance frequencies). Note that the outlier also shows a dip in one of these resonance frequencies, although its original spectrum does not have a peak there. Can you explain why it appears here?