Universität Ulm

Timo Tonn
Tobias Born
Summer term 2023
Points: 40
Submit until: 27.04.2023

Exercise Sheet 1 to NMDS

1. [**Image compression with singular value decomposition**, 0.5+2+3+3 points]

To start slowly, we begin with an application of the SVD. Therefore, recall its definition.

Theorem 1 (SVD, cf. lecture notes *Theorem 3.1.2*)**.** *Let $A \in \mathbb{R}^{m\times n}$ with rank r. There exist orthogonal matrices $U = \begin{bmatrix}u_1, \dots, u_m\end{bmatrix} \in \mathbb{R}^{m\times m}$ and $V = \begin{bmatrix}v_1, \dots, v_n\end{bmatrix} \in \mathbb{R}^{n\times n}$, such that*

$$U^T AV = \Sigma \in \mathbb{R}^{m\times n},$$

where Σ is a diagonal-shape matrix, i.e.,

$$\Sigma = \begin{cases} \begin{pmatrix} \hat{\Sigma} \\ 0 \end{pmatrix}, & \text{if } m \geq n, \\ \begin{pmatrix} \hat{\Sigma} & 0 \end{pmatrix}, & \text{if } m \leq n, \end{cases}$$

where $\hat{\Sigma} = \operatorname{diag}(\sigma_1, \dots, \sigma_p) \in \mathbb{R}^{p\times p}$ is diagonal, $p = \min(m, n)$ and

$$\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r > \sigma_{r+1} = \ldots = \sigma_p = 0.$$

The numbers $\sigma_1, \dots, \sigma_r > 0$ are called singular values of A.

A jpg-image consists of $m \times n$ pixels, from which each has a color represented as *rgb-value*. So a jpg-image corresponds to three $(m \times n)$-matrices A^r, A^g, A^b, one for the red, green and blue values. If one wants to reduce the image's storage size, one can use the SVD in the following way:

- For each of the three matrices, compute its SVD (e.g. $A^r = U^r\Sigma^r(V^r)^T$).
- Choose $k < \min(m, n)$ and compute the matrices A^r_k, A^g_k, A^b_k, where $A^r_k = U^r_k\Sigma^r_k(V^r_k)^T$ and $U^r_k = \begin{bmatrix}u^r_1, \dots, u^r_k\end{bmatrix}$, $V^r_k = \begin{bmatrix}v^r_1, \dots, v^r_k\end{bmatrix}$, $\Sigma^r_k = \operatorname{diag}(\sigma^r_1, \dots, \sigma^r_k)$.
- Create a new jpg-image from the matrices A^r_k, A^g_k, A^b_k.

a) Make yourself clear, how many pixels the compressed image will have.

b) What do you gain from performing this procedure? (Hint: storage size)

c) If a matrix is not regular, how can $k < \min(m, n)$ be chosen such that no information is lost? Does every such k yield an advantageous storage size?

d) Load *blumenwiese.jpg* into MATLAB and compress the image using different values for k. To compute an SVD, use the command *svd()*. For the conversions use the following code snippet.

```
image = double(imread('blumenwiese.jpg'));
Ar = image(:,:,1);
Ag = image(:,:,2);
Ab = image(:,:,3);

...

image(:,:,1) = Ark;
image(:,:,2) = Agk;
image(:,:,3) = Abk;
imshow(uint8(image));
```

Remark 1. *Choosing A_k^r, A_k^g and A_k^b like this has a deeper meaning. The matrix $A_k = U_k \Sigma_k V_k^T$, defined analogously to A_k^r, is the best approximation of $A \in \mathbb{R}^{m\times n}$ among all matrices with rank less or equal to k. This (and what best approximation means) will be discussed in detail in the next lectures.*

2. [**Important statements about the SVD**, 2.5+2.5+2.5 points]

The following statements are very important and will, among others, be crucially for the derivation of the algorithm for the SVD.

a) What is the connection between the SVD of A and the SVD of A^T?

b) What is the connection between the SVD of A and the eigenvalue decomposition of A^TA resp. AA^T? What can you say about the number of non-zero eigenvalues?

c) Show that $A = U\Sigma V^T \;\Leftrightarrow\; A = \sum_{i=1}^r \sigma_i u_i v_i^T$. This alternative representation of the SVD is called a *sum of rank-1 matrices*. Why?

3. [**Theorem 3.1.4 iii-v**, 3+3+3 points]

Now we will look at some more statements about the SVD from the lecture notes, for which the lecture notes lack a proof. Show that

a) $Av_i = \sigma_i u_i$ and $u_i^T A = \sigma_i v_i^T \;\forall i = 1, \dots, \min(m,n)$

b) $\mathrm{range}(A) = \mathrm{span}(u_1, \dots, u_r)$

c) $\ker(A) = \mathrm{span}(v_{r+1}, \dots, v_n)$

4. [**Bidiagonalization**, 2+4+7+2 points]

In the next lectures you will get to know the famous QR-algorithm, invented in 1965 by Golub, Kahan and Reinsch, to compute the SVD. It is quite sophisticated and therefore we will work it out over several exercise sheets. In this exercise sheet we will deal with an initial step, the bidiagonalization of a general real matrix.
For $A \in \mathbb{R}^{m\times n}$ we assume that $m \geq n$ (because of the connection between the SVD of A and A^T, cf. task 2a) and look for orthogonal matrices U_B and V_B such that

$$U_B^T A V_B = \begin{pmatrix} B \\ 0 \end{pmatrix} \in \mathbb{R}^{m\times n} \quad \text{and} \quad B = \begin{pmatrix} d_1 & f_1 & & 0 \\ 0 & \ddots & \ddots & \\ & & \ddots & f_{n-1} \\ 0 & & & d_n \end{pmatrix} \in \mathbb{R}^{n\times n}$$

For this, one needs so-called *Householder reflections.*

Theorem 2 (Householder reflections)**.** *For $x \in \mathbb{R}^n$, the Householder reflection $H \in \mathbb{R}^{n\times n}$ yields*

$$Hx = \mp \|x\|_2 e_1 \ ,$$

where $e_1 \in \mathbb{R}^n$ is the first canonical unit vector. H is defined as (I is the unit matrix)

$$H = I - \frac{2}{v^Tv} vv^T =: I - \beta vv^T \quad \textit{with} \quad v = x \pm \|x\|_2 e_1 \in \mathbb{R}^n \ .$$

The sign of v is chosen such that the procedure is numerically stable.

With this tool the bidiagonalization is quite straightforward. A sequence of multiplications with Householder reflections U_i and V_i yields

$$\begin{pmatrix} B \\ 0 \end{pmatrix} = U_n^T \cdots U_2^T U_1^T A V_1 V_2 \cdots V_{n-2} =: U_B^T A V_B^T.$$

We will clarify later, how U_i and V_i are chosen. Since Householder reflections are orthogonal matrices, U_B and V_B are also orthogonal.

a) *householder.m* computes a Householder reflection of $x \in \mathbb{R}^n$. To be numerically efficient, the definition of the Householder reflections is reformulated in a numerically more clever way and also the choice of the sign is realized. So make yourself familiar with the code, but you do not have to understand every step completely.

b) The bidiagonalization procedure (including the choice of the Householder reflections) is described in chapter 3.3.1 in the lecture notes. Read this chapter up to including Example 3.3.1 and make sure you understand it.

c) Create a MATLAB-program *bidiagonalization.m* that calculates the bidiagonalization of a matrix $A \in \mathbb{R}^{m\times n}$ with $m \geq n$. Test if

- the bidiagonalization is correct,
- U_B and V_B are orthogonal,
- A and B have the same singular values.

Test it for the cases $m = n$, $m > n$ and $m >> n$.

d) Now the question arises, why we bidiagonalize A instead of directly diagonalizing it. If we would do so, we would have computed the SVD. Unfortunately, this is not possible. Make yourself clear why.
For this, only apply the two first Householder reflections, namely the first from the left and the first from the right. Then do the same again, but this time try to also set $A(1, 2)$ to zero (what would be necessary to diagonalize A). With the command *spy(abs(A) > 1000*eps)* track which entries of A are zero. Which problem arises?

Remark 2. *There are a lot of algorithms to compute the SVD. Most of them include an initial bidiagonalization. The algorithm by Golub, Kahan and Reinsch is not the most efficient one, but it was one of the first ones and is probably the most famous.*
A direct diagonalization is in fact possible for a matrix with at most four singular values (but with other tools than Householder reflections). There are statements from the Galois theory which prohibit it in case of a matrix with more than four singular values.