Technische Universität München
Fakultät für Informatik
PD Dr. Tobias Lasser
Rüdiger Göbl
Anca Stefanoiu
Walter Simson

WiSe 2017
Assignment 4
November 21, 2017

Assignment for Basic Mathematical Tools

Exercise 1 Eigenvalues

For each of the following matrices, calculate all eigenvalues and the eigenvector corresponding to the largest (dominant) eigenvalue. Use the characteristic polynomial

$$\det(A - \lambda I) = 0$$

and the properties of the determinant where applicable ($\rightsquigarrow$ block matrices!).

Afterwards, implement the QR-Algorithm and compare your previously computed results with the algorithm's output.

a) $\begin{pmatrix} 0.5 & -2.5 \\ -2.5 & 0.5 \end{pmatrix}$

b) $\begin{pmatrix} 18 & -8 & -11 \\ 14 & -5 & -10 \\ 16 & -8 & -9 \end{pmatrix}$

c) $\begin{pmatrix} -3 & 1 & 0 & 0 & 0 \\ 0 & -2 & 0 & 0 & 0 \\ -8 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 3 & 2 \end{pmatrix}$

Exercise 2 Positive-definite matrices – Induced norm

Definition A matrix $A \in \mathbb{R}^{n\times n}$ is called *positive definite* if:

$$\langle \mathbf{x}, A\mathbf{x} \rangle = \mathbf{x}^\top A\mathbf{x} > 0 \quad \forall \mathbf{x} \in \mathbb{R}^n \backslash \{0\}$$

If the matrix is additionally symmetric, we say that A is *symmetric positive definite* (*spd*).

a) Consider a symmetric matrix $A \in \mathbb{R}^{n\times n}$. Show the following equivalence:

$$A \text{ spd} \;\Leftrightarrow\; \lambda_i > 0 \;\; \forall i \in 1, \ldots, n$$

With λ_i denoting the *eigenvalues* of A.

Hint A symmetric: All eigenvalues of A are real, A is diagonalizable ($\exists P \in \mathbb{R}^{n\times n}$ invertible, such that $P^{-1}AP$ is a diagonal matrix). Additionally, we know P to be orthogonal ($P^{-1} = P^\top$).

b) Now consider a spd matrix $A \in \mathbb{R}^{n\times n}$. Show that the following formula defines a *norm*:

$$\|\mathbf{x}\|_A = \sqrt{\langle \mathbf{x}, A\mathbf{x}\rangle}$$

This norm is often referred as *energy norm*.

Hint You may use that each scalar product induces a norm. Thus it is sufficient to show that $\langle \mathbf{x}, A\mathbf{x}\rangle$ with A spd indeed satisfies the properties of a scalar-product.

Exercise 3 (P) Image Compression

Consider the singular value decomposition of a real-valued matrix $A \in \mathbb{R}^{m\times n}$, i.e.

$$A = U\Sigma V^T,$$

where $U \in \mathbb{R}^{m\times m}$, $\Sigma \in \mathbb{R}^{m\times n}$, and $V \in \mathbb{R}^{n\times n}$. Assuming that the singular values are ordered, i.e., $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_p$ ($p = \min\{m,n\}$), a *low-rank* approximation of A can be obtained via

$$A = R_2 + R_2 + \ldots + R_k, \quad k \leq p,$$

where $R_i = \sigma_i u_i v_i^T$ (see `np.outer()`) and u_i and v_i denote the i-th column of U and V, respectively.

a) Write a Python script that loads an image of your choice, e.g., `tire.tif` in the project folder, and compute the singular value decomposition of the image.

b) Visualize some of the rank-one matrices R_i, e.g. the ones corresponding to $\sigma_1, \sigma_{10}, \sigma_p$. What do you observe?

c) Try to approximate the original image by summing up some rank-one matrices R_i and explain how such an approximation could be used for image compression.

Exercise 4 Eigenfaces

A popular application of an eigenvector decomposition is the so-called principal component analysis (PCA). Given a set of measurements vectors $x_1, x_2, \ldots, x_k \in \mathbb{R}^n$ whose entries may be considered as observations of possibly correlated (random) variables, PCA aims at turning these variables into linearly uncorrelated variables which are called principal components. This transformation is found by diagonalizing the covariance matrix $C = X^T X$, where

$$X = \left(x_1 - \bar{x} \middle| x_2 - \bar{x} \middle| x_3 - \bar{x} \middle| \ldots \middle| x_k - \bar{x} \right), \quad \bar{x} = \frac{1}{k}\sum_{i=1}^{k} x_i.$$

As C is symmetric, a diagonalization of C (obtained in Python by `[V,D] = np.linalg.eig(C)` for instance), yields a real-valued diagonal matrix

$$D = \begin{pmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_k \end{pmatrix} = V^T C V$$

and an orthogonal matrix V whose columns comprise the principal components. A particularly interesting application of PCA is face recognition. Given a large set of standardized pictures of human faces (features need to be aligned, same illumination, same resolution), we can treat those images as vectors of a high-dimensional feature space. Computing the principal components of that set of vectors, one obtains a basis for human faces (the *eigenfaces*), and it is possible to describe pictures of arbitrary faces as linear combination of eigenfaces with high accuracy. The coefficients can be used to identify identities.

a) Download and unzip the file `eigenfaces.zip` from the lecture homepage. It contains the files `eigenfaces.py` and face image directory `faces`.

b) The face images are loaded as "image vectors" into the matrix "faces". De-mean them as formulated above.

c) Compute the principal components of the set. Visualize that basis (as images perhaps?), showing the more significant basis vectors first.

d) Randomly select a couple of face images not used to compute the basis, project them into feature space, compress their representation[1], and compare them with the original images.

[1] This coefficient vector can be used as database identifier, and one could try to identify the person in other pictures based on identifier proximity.