MASD 2021, Assignment 3

Hand-in in groups of 2 or 3 before 30.9.2021 at 10.00
One submission per group
Remember to include the names of all group memebers

1 Identifying extremal points

Exercise 1 (**Extremal Points**). For each of the following functions, find all critical points and classify them as minimum, maximum, inflection point, or saddle. We use x_1 and x_2 to denote scalar parameters; if you find it easier you can rewrite the equations in matrix-vector notation. You are allowed to use library functions to help you with calculations (e.g., solve linear equations and find eigenvalues of matrices).

a) $f(x) = 2x^2 + (x-4)^3$

b) $f(x) = x^2 \ln x$

c) $f(x_1, x_2) = x_1^2 + 2x_1x_2 + 3x_2^2$

d) $f(x_1, x_2) = (x_1 - x_2)^2$

Deliverables. Derivations and the final answers. If you use library functions for computations, write down what function you used, including the input you gave to it and the output you got from it.

2 Gradient descent for the Netflix problem

Table 1 contains movie ratings ranging from 1 (bad) - 10 (good) for 10 different movies, for 6 different movie lovers. The "-" symbols represent movies that have not been rated because the viewer has not yet watched them. The *Netflix problem* is to predict the missing movie ratings in order to recommend movies that the viewers are likely to enjoy.

	Love Actually	Pride and Prejudice	Titanic	LaLa Land	Bridget Jones' Diary	Scream	Halloween	It	Sharknado 3	Pride, Prejudice and Zombies
Sophia	7	8	9	-	-	1	4	2	3	9
Anton	-	-	10	9	10	2	3	-	-	5
Fabio	10	9	-	8	-	-	-	2	1	3
Magda	1	-	2	-	-	9	8	9	-	-
Marietta	-	1	1	-	2	-	9	-	7	-
Carl	2	1	-	-	1	10	-	9	-	8

Table 1: A set of movie ratings with missing values corresponding to unseen movies.

In this exercise, we shall provide a solution to this problem using *matrix factorization*. We represent the movie rating table by a matrix M, where the "-" symbols are replaced by "0", as in (1), and seek two low-rank matrices A and B such that $A \times B \approx M$ in those entries that have data for M. We shall assume that A is a 6×2 matrix and that B is a 2×10 matrix, as follows:

$$M = \begin{pmatrix} 7 & 8 & 9 & 0 & 0 & 1 & 4 & 2 & 3 & 9 \\ 0 & 0 & 10 & 9 & 10 & 2 & 3 & 0 & 0 & 5 \\ 10 & 9 & 0 & 8 & 0 & 0 & 0 & 2 & 1 & 3 \\ 1 & 0 & 2 & 0 & 0 & 9 & 8 & 9 & 0 & 0 \\ 0 & 1 & 1 & 0 & 2 & 0 & 9 & 0 & 7 & 0 \\ 2 & 1 & 0 & 0 & 1 & 10 & 0 & 9 & 0 & 8 \end{pmatrix}, \quad A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ \vdots & \vdots \\ a_{61} & a_{62} \end{pmatrix}, \quad B = \begin{pmatrix} b_{11} & b_{12} & \dots & b_{1,10} \\ b_{21} & b_{22} & \dots & b_{2,10} \end{pmatrix} \tag{1}$$

We obtain such matrices A and B by solving the optimization problem

$$\text{argmin}_{A \in \mathbb{R}^{6\times 2}, B \in \mathbb{R}^{2 \times 10}} \|I \bigodot (M - AB)\|^2,$$

where $\bigodot$ denotes element-wise multiplication[1], and the matrix norm $\| \cdot \|$ is the norm (length) of the vector obtained by concatenating all the indices of the matrix $\cdot$ into a long vector[2]. Moreover, I is a binary 6×10 indicator matrix such that

$$\begin{aligned} I_{ij} &= 1 \quad \text{if } M_{ij} \neq 0, \\ I_{ij} &= 0 \quad \text{if } M_{ij} = 0 \end{aligned}$$

Exercise 2 (Netflix Part 1.). a) Explain in your own words what this model does, and how (if at all) we can interpret the two matrices A and B. It is completely fine to include thoughts that refer to your results in Exercise 4.

b) Show that, in coordinates, the error function simplifies as

$$E(A, B) = \|I \bigodot (M - AB)\|^2 = \sum_{i=1}^{6} \sum_{j=1}^{10} I_{ij} \left(M_{ij} - (a_{i1}b_{1j} + a_{i2}b_{2j})\right)^2$$

Deliverables. a) A short explanation, b) The derivation.

Exercise 3 (Netflix Part 2.). For any $k \in \{1, \dots, 6\}$, $l \in \{1, \dots, 10\}$, $m \in \{1, 2\}$ (specifying the indices of A and B), prove step by step that the following partial derivatives are correct:

$$\frac{\partial E}{\partial a_{km}} = 2 \sum_{j=1}^{10} I_{kj} \left(-M_{kj} b_{mj} + a_{k1} b_{1j} b_{mj} + a_{k2} b_{2j} b_{mj}\right)$$

$$\frac{\partial E}{\partial b_{ml}} = 2 \sum_{i=1}^{6} I_{il} \left(-M_{il} a_{im} + a_{i1} a_{im} b_{1l} + a_{i2} a_{im} b_{2l}\right)$$

Deliverables. The derivation.

Exercise 4 (Netflix Part 3.). a) Using these partial derivatives, fill in the Jupyter notebook template `A3template.ipynb`, Exercise 4, to implement a gradient descent algorithm[3] that minimizes E with respect to A and B. Provide a concise description of your implementation. Apply your implementation to the matrix M, which you find in the supplied file `netflix_matrix.txt`. Please include both your code and your final matrices A and B, as well as the matrix M' obtained by rounding all entries of AB to their nearest integer. How does M' compare with the original matrix M? Can you interpret the result?

b) What strengths and weaknesses do you see with this approach to the original Netflix problem? Do you have ideas for how it could be improved?

Deliverables. a) your code in the form of a filled-out Jupyter notebook template, a short implementation description, the three matrices, and a few lines discussion and interpretation; b) a few lines of discussion.

[1] **Hint:** The operation written as $\bigodot$ is performed by the numpy function `np.multiply`.

[2] The matrix norm can be computed using `np.linalg.norm`.

[3] **Hint:** You could concatenate the elements of A and B into a very long vector $(a_{11}, \dots, a_{62}, b_{11}, \dots b_{2,10})$ and compute gradients with respect to this. But since the gradient is defined element-wise, this is equivalent to simultaneously minimizing with respect to the separate matrices A and B, using the same learning rate for both and updating both in the same iteration. Note that the norm of this gradient is just the norm of the vector of all partial derivatives from c), concatenated. This can be implemented as `np.sqrt(np.norm(A)**2 + np.norm(B)**2)` (why?).