MASD 2020, Assignment 3

Hand-in in groups of 2 or 3 before 22.9.2020 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 extremal 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.
Solution:

a) We compute

$$f'(x) = 4x + 3(x-4)^2 = 4x + 3(x^2 - 8x + 16) = 3x^2 - 20x + 48.$$

Setting $f'(x) = 0$, we obtain (using the formula $x = \frac{-b \pm \sqrt{b^2-4ac}}{2a}$ to solve $ax^2 + bx + c = 0$):

$$3x^2 - 20x + 48 = 0 \Rightarrow x = \frac{20 \pm \sqrt{(-20)^2 - 4 \cdot 3 \cdot 48}}{2 \cdot 3} = \frac{20 \pm \sqrt{400 - 576}}{-6},$$

which gives two complex-valued solutions. As the function is defined only as a function of real numbers (that is the scope of the course), it thus does not have any critical points.

b) We compute

$$f'(x) = 2x \ln x + x^2 \cdot \frac{1}{x} = 2x \ln x + x = x(2 \ln x + 1).$$

Setting $f'(x) = 0$ to find critical points, we get

$$f'(x) = 0 \Rightarrow x(2 \ln x + 1) = 0.$$

This would imply either $x = 0$ or $2 \ln x + 1 = 0$, but note that the function f is not defined for $x = 0$ since $\ln 0$ is not defined. Thus, the only critical point is found when $2 \ln x + 1 = 0$, that is, $\ln x = -\frac{1}{2}$, which gives $x = e^{-\frac{1}{2}}$. In order to classify this critical point, let's compute the second derivatives of f:

$$f''(x) = x(\frac{2}{x}) + (2 \ln x + 1) = 3 + 2 \ln x.$$

Thus, we get $f''(e^{-\frac{1}{2}}) = 3 + 2 \ln(e^{-\frac{1}{2}}) = 3 + 2(-\frac{1}{2}) = 2 > 0$, which means that the critical point $x = e^{-\frac{1}{2}}$ is a minimum.

c) For a function of two variables $f(x_1, x_2)$, a critical point is a point (x_1, x_2) where $\nabla f(x_1, x_2) = 0$. We compute

$$\nabla f(x_1, x_2) = \begin{pmatrix} 2x_1 + 2x_2 \\ 2x_1 + 6x_2 \end{pmatrix} = \begin{pmatrix} 2 & 2 \\ 2 & 6 \end{pmatrix} \mathbf{x},$$

and see that $\nabla f(x_1, y_1) = \mathbf{0}$ if and only if:

$$\begin{aligned} 2x_1 + 2x_2 = 0 &\Leftrightarrow x_1 = -x_2, \text{ and} \\ 2x_1 + 6x_2 = 0 &\leftrightarrow x_1 = -3x_2. \end{aligned}$$

The only way both these equations can hold is if $x_1 = x_2 = 0$; thus $\mathbf{x} = \mathbf{0}$ is the only critical point of f. In order to classify it, we compute the Hessian

$$Hf(x_1, x_2) = \begin{pmatrix} 2 & 2 \\ 2 & 6 \end{pmatrix}.$$

The eigenvalues of the Hessian are (compute e.g. with numpy) 1.17 and 6.83; in particular both are positive, and hence the critical point $\mathbf{x} = \mathbf{0}$ is a minimum of f.

d) Again, to locate the critical points of f, we compute the gradient of f:

$$\nabla f(x_1, x_2) = \begin{pmatrix} 2x_1 - 2x_2 \\ -2x_1 + 2x_2 \end{pmatrix} = \begin{pmatrix} 2 & -2 \\ -2 & 2 \end{pmatrix} \mathbf{x}.$$

Setting $\nabla f(x_1, x_2) = 0$, we obtain:

$$\begin{aligned} 2x_1 - 2x_2 = 0 &\Leftrightarrow x_1 = x_2 \\ -2x_1 + 2x_2 = 0 &\Leftrightarrow x_1 = x_2. \end{aligned}$$

Thus, we find critical points of f along the entire line $x_1 = x_2$. To classify the critical points, we compute the Hessian of f:

$$Hf(x_1, x_2) == \begin{pmatrix} 2 & -2 \\ -2 & 2 \end{pmatrix},$$

which has eigenvalues 0 and 4. Since one eigenvalue is 0, the second derivative test is inconclusive. However, we note that along the line $x_1 = x_2$, the function has value $f(x_1, x_2) = 0$, while at any point (x_1, x_2) with $x_1 \neq x_2$, the function has a positive value $f(x_1, x_2) > 0$ (due to the power 2). Hence, every point along the line $x_1 = x_2$ is a minimum of f.

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.

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:

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

	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.

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.
Solution:

a) Each row of the matrix A corresponds to a person, and "scores" their preference for romantic versus horror movies, respectively. Each column of the matrix B corresponds to a movie, and "scores" the movie in the extent to which it is a romantic or horror movie, respectively.

b) We need to expand the matrix function $\|I \bigodot (M-AB)\|^2$ in terms of coordinate values for M, A, B and I.

[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`.

Let's start with the matrix product AB:

$$\begin{aligned} AB &= \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ \vdots & \vdots \\ a_{61} & a_{62} \end{pmatrix} \begin{pmatrix} b_{11} & b_{12} & \dots & b_{1,10} \\ b_{21} & b_{22} & \dots & b_{2,10} \end{pmatrix} \\ &= \begin{pmatrix} (a_{11}b_{11} + a_{12}b_{21}) & (a_{11}b_{12} + a_{12}b_{22}) & \dots & (a_{11}b_{1,10} + a_{12}b_{2,10}) \\ (a_{21}b_{11} + a_{22}b_{21}) & (a_{21}b_{12} + a_{22}b_{22}) & \dots & (a_{21}b_{1,10} + a_{22}b_{2,10}) \\ \vdots & \vdots & \ddots & \vdots \\ (a_{61}b_{11} + a_{62}b_{21}) & (a_{61}b_{12} + a_{62}b_{22}) & \dots & (a_{61}b_{1,10} + a_{62}b_{2,10}) \end{pmatrix}, \end{aligned}$$

whose $(i,j)^{th}$ entry is $a_{i1}b_{1j} + a_{i2}b_{2j}$. Thus, the $(i,j)^{th}$ entry of $M - AB$ must be $M_{ij} - (a_{i1}b_{1j} + a_{i2}b_{2j})$, and the $(i,j)^{th}$ entry of $I \bigodot (M - AB)$ is $I_{ij}(M_{ij} - (a_{i1}b_{1j} + a_{i2}b_{2j}))$. Hence, as the squared matrix norm is the sum of squared elements, we get

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

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.
Solution:
Since

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

we have

$$\frac{\partial E}{\partial a_{km}} = \frac{\partial}{\partial a_{km}}\sum_{i=1}^{6}\sum_{j=1}^{10} I_{ij}\left(M_{ij} - (a_{i1}b_{1j} + a_{i2}b_{2j})\right)^2 = \sum_{i=1}^{6}\sum_{j=1}^{10} I_{ij}\frac{\partial}{\partial a_{km}}\left(M_{ij} - (a_{i1}b_{1j} + a_{i2}b_{2j})\right)^2 \quad (2)$$

by the sum rule. Now,

$$\frac{\partial}{\partial a_{km}}\left(M_{ij} - (a_{i1}b_{1j} + a_{i2}b_{2j})\right)^2 = 2\left(M_{ij} - (a_{i1}b_{1j} + a_{i2}b_{2j})\right)\cdot\left(-\frac{\partial}{\partial a_{km}}(a_{i1}b_{1j} + a_{i2}b_{2j})\right) \quad (3)$$

by the chain rule. Note that

$$\begin{aligned} \tfrac{\partial}{\partial a_{km}}(a_{i1}b_{1j} + a_{i2}b_{2j}) &= \begin{cases} b_{1j} & \text{if } k = i, m = 1, \\ b_{2j} & \text{if } k = i, m = 2, \\ 0 & \text{if } k \neq i. \end{cases} \\ &= \begin{cases} b_{mj} & \text{if } k = i, \\ 0 & \text{if } k \neq i. \end{cases} \end{aligned} \quad (4)$$

Substituting (**??**) into (**??**), we obtain

$$\frac{\partial}{\partial a_{km}}\left(M_{ij} - (a_{i1}b_{1j} + a_{i2}b_{2j})\right)^2 = -2\left(M_{kj} - (a_{k1}b_{1j} + a_{k2}b_{2j})\right)\cdot b_{mj}, \quad (5)$$

which, substituting (??) into (??), gives

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

Next, we turn to $\frac{\partial E}{\partial b_{ml}}$, which is derived in a similar way:

$$\frac{\partial E}{\partial b_{ml}} = \sum_{i=1}^{6}\sum_{j=1}^{10} I_{ij}\frac{\partial}{\partial b_{ml}}\left(M_{ij} - (a_{i1}b_{1j} + a_{i2}b_{2j})\right)^2 \tag{6}$$

by the sum rule, and using the chain rule we obtain:

$$\begin{aligned}\tfrac{\partial}{\partial b_{ml}}\left(M_{ij} - (a_{i1}b_{1j} + a_{i2}b_{2j})\right)^2 &= 2\left(M_{ij} - (a_{i1}b_{1j} + a_{i2}b_{2j})\right)\tfrac{\partial}{\partial b_{ml}}\left(M_{ij} - (a_{i1}b_{1j} + a_{i2}b_{2j})\right)\\ &= \begin{cases} 2\left(M_{ij} - (a_{i1}b_{1j} + a_{i2}b_{2j})\right)a_{im} & \text{if } j = l,\\ 0 & \text{if } j \neq l.\end{cases}\end{aligned} \tag{7}$$

Substituting (??) into (??), we obtain

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

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.

Solution:

a) See the Jupyter notebook.

b) This approach has two main problems: First, there is no guarantee that the predicted scores lie between 1 and 10 – you can observe both negative scores and scores above 10. This could be alleviated using constrained optimization. Second, our approach models two clusters of movie types – but in general, we would not know how many types of movies are present in the dataset (and in general the clusters will appear based on viewer preference, not on genre). A higher number of clusters will result in a more precise model, but will also be computationally more demanding – so a problem is how to find a good compromise.

[3] **Hint:** You could concatenate the elements of A and B into a very long vector $(a_{11}, \ldots, a_{62}, b_{11}, \ldots b_{2,10})$ 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?).