Assignment 6

In the exercises 2–5 you are supposed to use `JAX` to set up the cost functions and gradients. In the exercises 6, 7, we employ the following notations: R_+ for the set of non-negative real numbers and R_{++} for the set of positive real numbers.

1. (**5**) Consider the function $f(x_1, x_2) = (x_1 + x_2^2)^2$. At the point $x = (1, 0)$, consider the search direction $p = (-1, 1)$. Show that this is a descent direction and find all minimizers of

$$\min_{\alpha > 0} f(x + \alpha p). \tag{1}$$

2. (**10**) Consider the example from the lecture: exponential function, uniformly sampled over the interval $[-3, 3]$:

$$x_i = -3 + 6\frac{i-1}{k-1}, \quad y_i = e^{x_i}, \quad i = 1, ..., k$$

with $k = 201$. Find the function of a form

$$f(x) = \frac{a_0 + a_1 x + a_2 x^2}{1 + b_1 x + b_2 x^2}$$

which minimizes $\max_{i=1}^{k} |f(x_i) - y_i|$. Find optimal values of a_0, a_1, a_2, b_1, b_2, and give the optimal objective value. Plot the data and the optimal rational function fit on the same plot. Since the `max`-function is not differentiable, consider replacing it by some differentiable approximation (it may help that `max` is equivalent to ∞-norm).

3. (**20***) Let us reconsider the Problem 3.7. Our original approach to the problem involved linearization of the defining equation

$$\theta_{123} = \arccos\frac{(\bar{r}_3 - \bar{r}_1) \cdot (\bar{r}_2 - \bar{r}_1)}{|\bar{r}_3 - \bar{r}_1||\bar{r}_2 - \bar{r}_1|}, \tag{2}$$

We are now equipped to reconsider this problem: using the same datasets as before, compute the best positions updates exactly, without assuming their smallness. To be more precise, you have to find the update with minimal possible L_2 norm among those satisfying the Eq. (2) exactly. When setting up the constraint function, try to avoid using `for` loops.

4. (**15**) Eigenvector u_0, corresponding to the smallest eigenvalue of a given symmetric matrix A can be found as a solution to the following constrained optimization problem:

$$u_0 = \mathrm{argmin}_{u \in R^d,\ |u|=1} u^T A u. \tag{3}$$

Assuming all eigenvalues of the matrix A are distinct, propose a generalization of this method to compute (simultaneously) two eigenvectors, corresponding to two lowest eigenvalues. Illustrate the validity of the proposed technique considering a random (symmetric) 10×10 matrix.

5. (**15**) Consider n sensors, located an unknown positions $x_1, ..., x_n$ in R^d. The goal is to locate the sensors, that is, to estimate the positions x_i, based on directional measurements. For each pair of sensors (i, j) we obtain a noisy measurement of the direction from x_j to x_i:

$$n_{ij} \approx \frac{x_i - x_j}{|x_i - x_j|}. \tag{4}$$

In principle, the task of estimation of the sensors' positions can be written as optimization problem

$$\min_{x_1,...,x_n \in R^d} \sum_{(i,j)} \| (x_i - x_j) - n_{ij}\left((x_i - x_j) \cdot n_{ij}\right) \|_2. \tag{5}$$

This task is, however, ambiguous: clearly, the directional measurements determine the positions only up to i) global translation of all sensors and ii) rescaling of all the coordinates arbitrarily. Your task is to find *any* set of coordinates among those delivering the minima in Eq. (5) for the data given in `data_sensors.pickle` [pickled dict of the form `{#: (`i`, `j`, `n_{ij}`)}`]. You should be careful, since a blind minimization of the cost function in Eq. (5) will deliver all $x_i = 0$ which is, of course, meaningless.

6. (**4**) Check if the following sets are convex:
   (a) (**1**) $\left\{(x,y) \in R^2_{++} | x/y \le 1\right\}$
   (b) (**1**) $\left\{(x,y) \in R^2_{++} | x/y \ge 1\right\}$
   (c) (**1**) $\left\{(x,y) \in R^2_{++} | xy \le 1\right\}$
   (d) (**1**) $\left\{(x,y) \in R^2_{++} | xy \ge 1\right\}$

7. (**14**) Using the second-order conditions, show that:
   (a) (**2**) the function $f(x,y) = x^2/y$ defined on $y > 0$ is convex.
   (b) (**4**) the geometric mean $f(x) = (\Pi_{k=1}^n x_k)^{1/n}$ defined on R^n_{++} is concave.
   (c) (**4**) the function $f(x) = \ln \sum_{k=1}^n \exp(x_k)$ defined on R^n is convex.
   (d) (**4**) the function $f(x,t) = -\ln\left(t^2 - x^T x\right)$ defined on $\{(x,t) \in R^n \times R | t > \|x\|_2\}$ is convex.

8. (**8**) Show that the following functions of vector x are convex
   (a) (**4**) $f(x) = \frac{\|Ax-b\|_2^2}{1-x^T x}$ on $\{x | \|x\|_2 < 1\}$.
   (b) (**4**) $f(x) = \begin{cases} \frac{1}{2}\|x\|_2^2, & \|x\|_2 \le 1 \\ \|x\|_2 - \frac{1}{2}, & \|x\|_2 > 1 \end{cases}$.

9. (**6**) In this exercise, we will consider three measures of the spread of a group of numbers. For $x \in R^n$, define three functions:
   (a) (**2**) spread of the data:
   $$f_1(x) = \max_i x_i - \min_i x_i, \tag{6}$$
   (b) (**2**) standard deviation of the data:
   $$f_2(x) = \left(\frac{1}{n}\sum_{i=1}^n x_i^2 - \left(\frac{1}{n}\sum_{i=1}^n x_i\right)^2\right)^{1/2}, \tag{7}$$
   (c) (**2**) average absolute deviation from the median of the data $\mathrm{med}(x)$:
   $$f_3(x) = \frac{1}{n}\sum_{i=1}^n |x_i - \mathrm{med}(x)|. \tag{8}$$

   Which of these three functions is convex? For each one, either show that it is convex, or give a counterexample showing it is not convex.

10. (**10**) In the file `data_fit.npz`, you will find a set of values x and y. Your task is to find two fits of the dependence $y_i \sim f(x_i)$ with the function
    $$f(x|a,b,c) = ax^2 + bx + c \tag{9}$$
    which minimize i) the sum of the errors squared $\sum_i (y_i - f(x_i))^2$ and ii) the sum of the absolute errors $\sum_i |y_i - f(x_i)|$. For the first case, compute the solution using both `numpy.linalg.lstsq` and `cvxpy` (the results should be the same, of course) and for the second case, consider solution via `cvxpy` only. Plot two fits together with the original points. Which fit do you find more visually appealing and why did it turn out better?

11. (**10**) In the file `data_linprog.npz`, you will find a set of n values y and n vectors $x_i \in R^k$ (the latter are arranged into a matrix $X \in R^{n\times k}$). Since $n < k$, there exist many vectors $a \in R^k$ such that $y_i = a \cdot x_i$ for all i. Your task is to find among them the vector with the smallest L_1 norm. Reduce this problem to the linear program (consult the Lecture 7 if needed) and solve this linear program via `scipy.optimize.linprog` [be careful with the `bounds` argument which is not innocent by default]. How many non-zero elements does the optimal a have?

12. (**20**) In the file `data_regr.npz`, you will find a set of n values y and n vectors $x_i \in R^k$ (the latter are arranged into a matrix $X \in R^{n\times k}$). The dependence between y and x is linear:

$$y_i = a \cdot x_i + \epsilon_i \tag{10}$$

where $a \in R^k$ and ϵ_i is random noise. In fact, some of the variables x_i are redundant, meaning they don't have *any* impact on the values of y_i (corresponding entries of the vector a are zeroes). The one who generated the data knows the underlying vector a, and your task is to recover it, using the observations X and y. In the problem of this sort one usually requires a to be small in some sense. Overall, it takes to minimize:

$$L_n(a) = \sum_i (y_i - a \cdot x_i)^2 + \lambda \|a\|_n, \tag{11}$$

where λ is a hyperparameter (meaning it is the user of Eq. (11) who decides on its value: consider several values of λ which make sense to you). In this problem, you have to consider $n = 2$ and $n = 1$ and use `cvxpy` to solve the optimization problem. Which of the variables x_i are actually not relevant for predicting y?

13. (**40**) A signal emitted by a source at an unknown position $r \in R^2$ is received by m sensors at known positions $r_1, ..., r_m \in R^2$ (see `P.npy`). From the strength of the received signals (see `d.npy`), the noisy estimates d_k of the distances $\|r - r_k\|_2$ are obtained. We are interested in estimating the source position r based on the measured distances d_k. Formally, we have to minimize:

$$L(r) = \sum_k \left(\|r - r_k\|_2^2 - d_k^2\right)^2 \tag{12}$$

over r at given r_k and d_k.

- (**5**) Is the function in Eq. (12) convex?
- (**25**) Your goal in this task is to find the global minimum using the method of Lagrange multipliers. Consider rewriting the optimization problem identically to

$$\min_{r,t} \sum_k \left(t + \|r_k\|_2^2 - 2r \cdot r_k - d_k^2\right)^2 \quad \text{s.t.} \quad \|r\|_2^2 = t \tag{13}$$

  This minimization under constraint is reduced to finding stationary points in $r = (x, y)$ and t of the following Lagrangian

$$\bar{L}(x, y, t) = \sum_k \left(t + \|r_k\|_2^2 - 2r \cdot r_k - d_k^2\right)^2 + \lambda \left(\|r\|_2^2 - t\right). \tag{14}$$

  Since this is just a quadratic form over x, y, t, the stationary point can be found explicitly as $x(\lambda), y(\lambda), t(\lambda)$ (express these functions in terms of r_k and d_k). NB: `sympy` can be useful at some point.
- (**10**) Solve numerically the polynomial equation $x(\lambda)^2 + y(\lambda)^2 - t(\lambda) = 0$ over λ to find the point, which delivers the global minimum for the given data. Plot the graph with (i) positions of the detectors, (ii) contour lines of $L(r)$, (iii) locally extremal point(s) and (iv) the global minimum. Characterize the locally extremal point (local maximum/minimum/saddle point).

14. (**25**) Consider trading a universe of n stocks over T days, with noisy predictions for the stock returns p_{ti} available in `pred.npy` (the rows are days, the data start with day 0). Assuming that we have the position evolving as π_t (π_t at each t is a vector of n components), expected risk-adjusted gain G reads

$$G = \sum_t \left[p_t \cdot \pi_t - \pi_t^T \cdot \Omega \cdot \pi_t - \gamma \sum_i |\pi_{t,i} - \pi_{t-1,i}| \right] \tag{15}$$

and has to be maximized over π_t (pick $\gamma = 0.01$). The matrix Ω is available in `cov.npy`.

- (**5**) Start with $t = 1$ ($\pi_0 = 0$ by definition). At this moment, you have access to p_1 and have to maximize:

$$p_1\pi_1 - \pi_1\Omega\pi_1 - \gamma\sum_i |\pi_{1,i} - 0| \longrightarrow \max$$

  over π_1. Show that this is a concave function of π_1 and maximize it using `cvxpy`.

- (**10**) At $t = 2$ you already know π_1. At this moment, you have access to p_2 and have to maximize:

$$p_2\pi_2 - \pi_2\Omega\pi_2 - \gamma\sum_i |\pi_{2,i} - \pi_{1,i}| \longrightarrow \max$$

  Repeat the process until you reach the end of the time series. The corresponding π_{ti} should be stored as a file: this will be you 1st result in this problem.

- (**5**) considering the case of $\gamma = 0$: in this case, optimization of G can be done directly. Make sure the result of such direct computation coincides with `cvxpy` result. The corresponding π_{ti} should be stored as a file: this will be your 2nd result in this problem.

- (**5**) Compute expected gain/costs (the 1st and the 3rd terms in Eq. (15)) over the full period for two trading strategies computed above (note that the trading costs are present even if you decided to optimize at $\gamma = 0$).

Interpretation. In this problem, you are given the noisy estimates of the stock returns, p_{it} and the true returns r_{it} are known only to the problem author. If the true returns are known, the trading strategy (π_t) can be evaluated by computing the PnL:

$$\mathrm{PnL}_t = r_t \cdot \pi_t - \gamma\sum_i |\pi_{t,i} - \pi_{t-1,i}| \tag{16}$$

After the problem is solved, we will compute PnL_t and discuss the results.