Exercises for Lecture Course on Numerical Optimization (NUMOPT)
Albert-Ludwigs-Universität Freiburg – Winter Term 2019-2020

Exercise 2: Duality, Semidefinite Programming and Fitting Problems

Prof. Dr. Moritz Diehl, Dimitris Kouzoupis and Florian Messerer

The goal of this exercise is to first train the derivation of dual problems and then to explore the potential of Semidefinite programming by means of a practical example. The aim of the last exercises is to familiarize with linear least squares fitting problems.

Exercise Tasks

1. **Lagrange duality and dual problems:**

   (a) Derive the explicit form of the dual of the following *logarithmic barrier* problem:

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & c^T x - \sum_{j=1}^{n} \log x_j \\ \text{s.t.} \quad & a^T x = b, \end{aligned}$$

   where $a, c \in \mathbb{R}^n$ and b is a scalar.
   Remark: Problems using a logarithmic barrier as the one above will be at the core of interior point methods that we will analyze later in this course.

   (2 points)

   Solution: the Lagrangian of the problem reads

$$L(x, \lambda) := c^\top x - \sum_{j=1}^{n} \log x_j - \lambda(a^\top x - b) = (c^\top - \lambda a^\top)x - \sum_{j=1}^{n} \log x_j + \lambda b.$$

   In order to derive the dual, we have to find a solution to the following problem:

$$\min_{x \in \mathcal{D}} \quad L(x, \lambda),$$

   where $\mathcal{D}$ represents the domain of the functions appearing in the primal. If $c_j - \lambda a_j \leq 0$ for some component j, the Lagrangian is unbounded, so we will impose the condition

$$c_j - \lambda a_j > 0, \quad \forall j = 0, \ldots, n.$$

   Taking the derivative with respect to x, we have

$$\nabla_x L(x, \lambda) = c - X^{-1}\mathbf{1} - \lambda a,$$

   with $X = \text{diag}(x)$, The j-th component of gradient of the Lagrangian vanishes for

$$x_j^* = \frac{1}{c_j - \lambda a_j},$$

   where, in order to derive such condition, we have to impose $x_j > 0$, which is anyway required by the domain conditions on the logarithm in the objective of the primal problem. As $L(x, \lambda)$ is convex in x, such conditions attain the global minimum of the Lagrangian over $\mathcal{D}$. Plugging the minimizer in the Lagrangian we obtain the dual problem:

$$\begin{aligned} \min_{\lambda \in \mathbb{R}} \quad & c^T x^* - \sum_{j=1}^{n} \log x_j^* + \lambda(a^T x^* - b) \\ \text{s.t.} \quad & c - \lambda a > 0, \end{aligned}$$

(b) Consider the following *mixed-integer quadratic program* (MIQP):

$$\begin{aligned}\min_{x\in\{0,1\}^n} \quad & x^TQx + q^Tx \\ \text{s.t.} \quad & Ax \geq b,\end{aligned}$$

where the optimization variables x_i are restricted to take values in $\{0, 1\}$. Solving mixed-integer problems is in general a challenging task, thus it is common practice to exploit continuous reformulations as the following:

$$\begin{aligned}\min_{x\in\mathbb{R}^n} \quad & x^TQx + q^Tx \\ \text{s.t.} \quad & Ax \geq b \\ & x_i(1-x_i) = 0 \qquad i = 0, \cdots, n-1.\end{aligned}$$

i. Is this reformulation convex? (1 point)

   Solution: no, it has nonlinear equality constraints, hence is not convex.

ii. A lower bound to the optimal solution can be computed by solving the (convex) dual problem (not required here). Derive the explicit form of the dual of the continuous reformulation.

   (2 points)

   Solution: the Lagrangian of the problem reads

$$L(x, \lambda, \nu) = x^TQx + q^Tx - \nu^T(Ax - b) - \lambda^TX(1 - x),$$

   which is unbounded if $Q + \Lambda \prec 0$ or $Q + \Lambda \succeq 0$ and $q - \lambda - A^T\nu \in \mathcal{N}(Q + \Lambda)$ (with $\Lambda = \operatorname{diag}\lambda$). The minimum is attained for

$$x^* = -\frac{1}{2}(Q + \Lambda)^+(q - \lambda - A^T\nu).$$

   Plugging this in the Lagrangian and adding the conditions obtained for the dual function to be bounded, we obtain the following dual problem:

$$\begin{aligned}\min_{x\in\mathbb{R}^n} \quad & x^{*T}Qx^* + q^Tx^* - \nu^T(Ax^* - b) - \lambda^TX^*(1 - x^*) \\ \text{s.t.} \quad & Q + \Lambda \succeq 0 \\ & q - \lambda - A^T\nu \in \mathcal{R}(Q + \Lambda) \\ & \nu \geq 0.\end{aligned}$$

2. **Regularized linear least squares:** Given a matrix $J \in \mathbb{R}^{m \times n}$ with arbitrary dimensions, a symmetric positive definite matrix $Q \succ 0$, a vector of measurements $\eta \in \mathbb{R}^m$ and a point $\bar{x} \in \mathbb{R}^n$, calculate the limit:

$$\lim_{\substack{\alpha \to 0 \\ \alpha > 0}} \arg\min_x \frac{1}{2}\|\eta - Jx\|_2^2 + \frac{\alpha}{2}(x - \bar{x})^\top Q(x - \bar{x}). \tag{1}$$

Hint: Use matrix square root and the Moore–Penrose pseudoinverse, i.e., SVD of a suitable matrix.

(3 points)

Q is symmetric positive definite and therefore has unique square root $Q^{\frac{1}{2}}$, such that $Q = Q^{\frac{1}{2}\top} Q^{\frac{1}{2}}$

$\frac{1}{2}\|\eta - Jx\|_2^2 + \frac{\alpha}{2}(x - \bar{x})^\top Q(x - \bar{x})$

$= \frac{1}{2}\|\eta - Jx\|_2^2 + \frac{\alpha}{2}(x - \bar{x})^\top Q^{\frac{1}{2}\top} Q^{\frac{1}{2}}(x - \bar{x}) = \frac{1}{2}\|\eta - Jx\|_2^2 + \frac{\alpha}{2}\|\underbrace{Q^{\frac{1}{2}}(x - \bar{x})}_{=:y}\|_2^2 = \dots$

Substitute $y := Q^{\frac{1}{2}}(x - \bar{x}) \Leftrightarrow x = Q^{-\frac{1}{2}}y + \bar{x}$

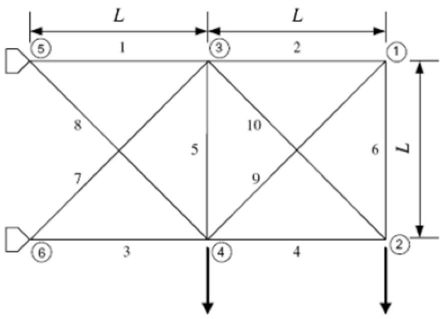
$\dots = \frac{1}{2}\|\underbrace{\eta - J\bar{x}}_{\widetilde{\eta}} - \underbrace{JQ^{-\frac{1}{2}}}_{\widetilde{J}}y\|_2^2 + \frac{\alpha}{2}\|y\|_2^2 = \frac{1}{2}\|\widetilde{\eta} - \widetilde{J}y\|_2^2 + \frac{\alpha}{2}\|y\|_2^2$

This is now the same form as problem (6.20) in the lecture notes (p. 46), and to obtain the limit in a clean way we can follow the steps outlined in the proof of Lemma 6.1.

$$\lim_{\substack{\alpha \to 0 \\ \alpha > 0}} \arg\min_y \frac{1}{2}\|\widetilde{\eta} - \widetilde{J}y\|_2^2 + \frac{\alpha}{2}\|y\|_2^2 = \lim_{\substack{\alpha \to 0 \\ \alpha > 0}} (\widetilde{J}^\top \widetilde{J} + \alpha I)^{-1}\widetilde{J}^\top \widetilde{\eta} \overset{\text{Lem. 6.1}}{=} J^\dagger \widetilde{\eta}$$

with $\widetilde{J}^\dagger$ the Moore Penrose Pseudo inverse. So in the limit we obtain $y^* = J^\dagger \widetilde{\eta}$, and substituting back, $x^* = Q^{-\frac{1}{2}}y^* + \bar{x}$.

3. **Truss design:** Aim of this task is to design a truss topology with minimum compliance under the influence of external forces. Assume we have the following structure with 6 nodes and 10 bars



with external forces f_e. Under these forces, the nodes are displaced on the directions they are free to move until a certain equilibrium is reached. We denote the vector of displacements with u. Our goal is to find the optimal cross-sectional area x_i of each bar i that minimizes the compliance $f_e^\top u$ of the structure while respecting restrictions on the available material. The reaction forces that are caused by the external load depend linearly on u via the stiffness matrix $K(x)$, i.e., $f_r = -K(x)u$. On the other hand, at equilibrium, it should also hold $f_r = -f_e$. Taking also into account the constraints on the materials, we end up with the following optimization problem:

$$\underset{u,x}{\text{minimize}} \; f_e^\top u \tag{2a}$$

$$\text{subject to: } K(x)u = f_e \tag{2b}$$

$$0 \leq x_i \leq x_{\max} \tag{2c}$$

$$\sum_{i=1}^{m} l_i x_i \leq V_{\max} \tag{2d}$$

where m is the number of bars, $x_{\max}$ the maximum cross-sectional area, l_i the length of bar i and $V_{\max}$ the maximum allowed volume for the structure. The stiffness matrix depends linearly on the cross-sectional area of each bar via the relation $K(x) = \sum_{i=1}^{m} K_i x_i$.

At first glance, Problem (2) seems like a hard, highly nonlinear problem. However, after some manipulation, we can derive an equivalent convex problem in a form suitable for an SDP solver, namely:

$$\underset{x,\,\alpha}{\text{minimize}} \quad \alpha \tag{3a}$$

$$\text{subject to:} \quad \begin{bmatrix} \alpha & f_e^\top \\ f_e & K(x) \end{bmatrix} \succeq 0 \tag{3b}$$

$$\text{Constraints (2c) and (2d)} \tag{3c}$$

(a) Show how problem (2) can be transformed to the equivalent problem (3). You will need to use transformations similar to the previous exercise sheet as well as the Schur complement. Keep in mind that matrix $K(x)$ is strictly positive definite for stable structures.

(2 points)

$$\begin{array}{ll} \min\limits_{u,x} & f_e^\top u \\ \text{s.t.} & K(x)u = f_e \end{array} \Leftrightarrow \min_x \quad f_e^\top K^{-1}(x) f_e \Leftrightarrow \begin{array}{ll} \min\limits_{\alpha,x} & \alpha \\ \text{s.t.} & \alpha \geq f_e^\top K^{-1}(x) f_e \end{array}$$

Schur complement: $X = \begin{bmatrix} A & B \\ B^\top & C \end{bmatrix}$. Then if $C \succ 0$ we have

$$X \succeq 0 \Leftrightarrow A - BC^{-1}B^\top \succeq 0$$

(check, e.g., Boyd, Convex Optimization, Appendix A.5.5).
Identify $A = \alpha$, $B = f_e^\top$, $C = K(x) \succ 0$. Therefore

$$\begin{array}{ll} \min\limits_{x,\alpha} & \alpha \\ \text{s.t.} & \alpha - f_e^\top K^{-1}(x) f_e \geq 0 \end{array} \Leftrightarrow \begin{array}{ll} \min\limits_{x,\alpha} & \alpha \\ \text{s.t.} & \begin{bmatrix} \alpha & f_e^\top \\ f_e & K(x) \end{bmatrix} \succeq 0 \end{array}$$

(b) Solve problem (2) with CasADi and IPOPT using the provided template and functions. Take $x_{\max} = 200$ and $V_{\max} = 10^5$. Carefully decide how to initialize the decision variables. Not every initialization will work.

(2 points)

(c) At the moment CasADi does not support any SDP solver. To solve problem (3) we will use YALMIP instead. Similar to CasADi it provides a flexible syntax/symbolic framework for formulating optimization problems and passing them to solvers.

Download `https://github.com/yalmip/YALMIP/archive/master.zip` and unzip it to the MATLAB directory or any directory of your choice. In MATLAB navigate to this directory and run `>> addpath(genpath('yalmip-master'))` in the command line (adapt to foldername if necessary). Test the installation via `>> yalmiptest`. To save the path beyond your current MATLAB session, run `>> savepath`.

As SDP solver we will use SDPT3. Clone or download and extract the zipped folder from `https://github.com/SQLP/SDPT3` into the MATLAB directory or any directory of your choice. Run the file `install_sdpt3.m` to install followed by `>> savepath` to make the installation permanently available.

Solve Problem (3) with YALMIP and SDPT3 using the provided template and functions.

(3 points)

4. **Linear L_2 fitting:** Assume we have a set of N noisy measurements $(x_i, \tilde{y}_i) \in \mathbb{R}^2$ onto which we would like to fit a line $y = ax + b$. This task can be expressed by the optimization problem:

$$\min_{a,b} \sum_{i=1}^{N} (ax_i + b - \tilde{y}_i)^2 = \min_{a,b} \left\| J \begin{pmatrix} a \\ b \end{pmatrix} - \tilde{y} \right\|_2^2. \tag{4}$$

As discussed in the lecture, the optimal solution of (4) can be calculated explicitly by solving the linear system:

$$J^T J \begin{pmatrix} a \\ b \end{pmatrix} = J^T \tilde{y}, \tag{5}$$

(a) Generate the problem data. Take $N = 30$ points in the interval $[0, 5]$ and generate the true outputs $y_i = 3x_i + 4$. Add Gaussian noise with zero mean and standard deviation 1 to get the noisy measurements $\tilde{y}_i$ and plot the results. *Hint: lookup* `linspace` *and* `randn` *commands, e.g. via using* `help randn` *or* `doc randn` *in the command line. If you want a reproducible 'random' sequence, you can use* `rng`*.*

(1 point)

(b) Write down matrix J. Calculate the coefficients a, b in MATLAB using Equation (5) and plot the obtained line in the same graph as the measurements.

(2 points)

$J = \begin{bmatrix} x & 1 \end{bmatrix}$

(c) Introduce 3 outliers in your measurements y and plot the new fitted line in your plot.

(1 point)

(d) Solve question 2(b) with CasADi and compare the results.

(1 point)

You will need the measurements y (both with and without outliers) and the matrix J for the next task.

5. **Linear L_1 fitting:** In this task we want to fit a line to the same set of measurements, but we use a different cost function:

$$\min_{a,b} \sum_{i=1}^{N} |(ax_i + b - y_i)|. \tag{6}$$

This objective is not differentiable, so we will need auxiliary variables to form an equivalent problem. We introduce the so-called slack variables $s_1, \ldots, s_N$ and solve instead:

$$\min_{a,b,s} \sum_i s_i \tag{7a}$$

$$\text{s.t.} \; -s_i \leq ax_i + b - y_i \leq s_i, \qquad i = 1, \ldots, N, \tag{7b}$$

$$-s_i \leq 0, \qquad i = 1, \ldots, N. \tag{7c}$$

(a) Problem (7) is a Linear Program. In order to solve it with `linprog`, the native LP solver of MATLAB, we need to bring it to the form:

$$\min_z f^T z \tag{8a}$$

$$\text{s.t } Az \leq b \tag{8b}$$

$$Cz = d \tag{8c}$$

$$l_z \leq z \leq u_z, \tag{8d}$$

Write matrix A and vectors f, b on paper. Order your variables as $z = [a, b, s_1, \ldots, s_N]$. Use matrix J from the previous exercise to define A.

(2 points)

$$f = \begin{bmatrix} 0 \\ 0 \\ 1 \\ \vdots \\ 1 \end{bmatrix} A = \begin{bmatrix} J & -I \\ -J & -I \\ 0 & -I \end{bmatrix}, b = \begin{bmatrix} y \\ -y \\ 0 \end{bmatrix}$$

(b) Solve the problem using the measurements y from the previous exercise (both with and without outliers) and plot the results against those of the L2 fitting. Which norm performs better?

(1 point)

The L1 norm is more robust against the outliers (as it does not penalize the model-measurement-mismatch quadratically). Which norm performs better depends on the context, but here it seems like we want our method to 'ignore' the outliers (the outliers seem nonsensical). That means L1 performs better.

(c) Solve Problem (7) with CasADi and compare the results.

(1 point)

This sheet gives in total 24 points