UNIVERSITÄT BERN

Applied Optimization
Exercise 4 - Duality

October 20, 2021

Hand-in instructions:

Please hand-in **only one** compressed file named after the following convention: `Exercise`n`-GroupMemberNames.zip`, where n is the number of the current exercise sheet. This file should contain:

- The complete code folder except for the `build` subfolder. Make sure that your code submission compiles on your machine. Zip the code folder.
- A `readme.txt` file containing a description on how you solved each exercise (use the same numbers and titles) and the encountered problems.
- Other files that are required by your `readme.txt` file. For example, if you mention some screenshot images in `readme.txt`, these images need to be submitted too.
- Submit your solutions to ILIAS before the submission deadline.

Lagrange Duality (3 pts)

Consider the optimization problem

$$\begin{array}{ll} \text{minimize} & x^2+1 \\ \text{subject to} & (x-2)(x-4) \leq 0 \end{array}$$

with variable $x \in \mathbb{R}$.

(a) Analysis of primal problem: give the feasible set, the optimal value, and the optimal solution.

(b) Lagrangian and dual function: plot the objective x^2+1 versus x. On the same plot, show the feasible set, optimal point and value, and plot the Lagrangian $L(x,\lambda)$versus x for a few positive values of λ. Verify the lower bound property ($p^* \geq inf_x L(x,\lambda)$ for $\lambda \geq 0$). Compute and sketch the Lagrange dual function g.

(c) Lagrange dual problem: state the dual problem, and verify that it is a concave maximization problem. Find the dual optimal value and dual optimal solution λ. Does strong duality hold?

KKT Condition (2 pts)

Consider the following optimization problem:

$$\begin{array}{ll} \text{minimize} & x_1^2 - 2x_2^2 \\ \text{subject to} & (x_1+4)^2 - 2 \leq x_2 \\ & x_1 - x_2 + 4 = 0 \\ & x_1 \geq -10 \end{array}$$

(1) sketch the problem and graphically determine the primal solution x^*.

(2) verify your x^* by determining suitable λ^* and ν^* such that the KKT conditions are satisfied for (x^*, λ^*, ν^*).

Programming (5 pts)

For the following optimization problem in the standard form:

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & f_i(x) \leq 0, i = 1, ..., m \\ & h_i(x) = 0, i = 1, ..., p \end{array}$$

with variable $x \in \mathbb{R}^n$.

Implement an optimality checker to see if a query point x satisfies the KKT condition w.r.t. a user-specified tolerance ϵ. This ϵ tolerance is required because of floating-point precision limitations. Indeed, it might happen that a result should really be exactly zero but because of floating-point issues, it's not quite zero. e.g. instead of checking that $f_i(x) \leq 0$ you should check that $f_i(x) \leq \epsilon$.

To implement this optimality checker, you will have to implement the function `is_KKT_satisfied (...)` in the `include/Utils/OptimalityChecker.hh` file. The input of the function is the problem, query point and the corresponding λ and ν.

As a special case, you could test the optimality checker with the quadratically constrained quadratic program:

$$\begin{array}{ll} \text{minimize} & 1/2x^T A_0 x + b_0^T x + c_0 \\ \text{subject to} & 1/2x^T A_i x + b_i^T x + c_i \leq 0, i = 1, ..., m \\ & Cx = d \end{array}$$

with variable $x \in \mathbb{R}^n$, $C, A_i \in \mathbb{R}^{n \times n}$, $b_i \in \mathbb{R}^n$ and $c_i, d \in \mathbb{R}$. The class `FunctionQuadraticND` can be reused in this case. You thus need to implement the `eval_gradient(...)` function in `Include/Function/FunctionQuadraticND.hh`.

(more on the next page)

Your last task is to implement the minimization problem given above in the KKT Condition exercise. This should be done by filling the main function in `main.cc`, where you can setup the n dimension quadratic function by manually giving the A and b. Store the equality constraints and the inequality constraints, as well as λ and ν, in the corresponding vectors that are provided. You can look for the ToDos included in the code to help you.

Note that you can also run the `OptimalityChecker-test` executable to make sure your implementation passes all tests. Again, passing the tests is not a sufficient condition to obtain all points for this exercise.