

Solutions to hw4 homework on Convex
Optimization
<https://web.stanford.edu/class/ee364a/homework.html>

Andrei Keino

July 2, 2020

5.1

A simple example. 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 ($p^* \leq \inf_x L(x, \lambda)$ for $\lambda \geq 0$). Derive 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?

(d) Sensitivity analysis. Let $p^*(u)$ denote the optimal value of the problem

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

as a function of the parameter u . Plot $p^*(u)$. Verify that $dp^*(0)/du = -\lambda^*$.

Solution:

(a)

The feasible set is $x \in [2, 4]$. The optimal solution is $x^* = 2$, the optimal value is $p^* = 5$.

(b)

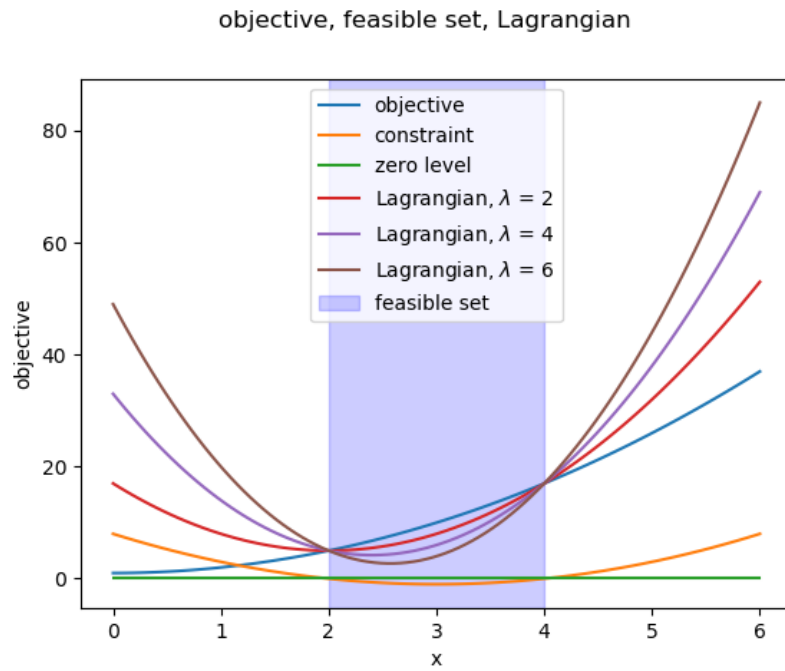


Figure 1: objective, feasible set, lagrangian for this problem.

It's easy to see from the Figure 1, that the Lagrangian values on the feasible set are less or equal than the objective values on the feasible set, i. e. $(p^* \geq \inf_x L(x, \lambda) \text{ for } \lambda \geq 0)$.

(c)

The dual objective function for this problem can be found solving the constrained equation for the Lagrangian:

$$g(\lambda) = \inf_x (x^2 + 1 + \lambda(x - 2)(x - 4))$$

subject to $\lambda \geq 0$

The Lagrangian reaches its minimum at the point $\tilde{x} = \frac{3\lambda}{\lambda+1}$. Then the dual objective itself is:

$$g(\lambda) = -\lambda + 10 - \frac{9}{(\lambda+1)}$$

The second derivative of the dual function is:

$$g''(\lambda) = -18/(\lambda+1)^3$$

which is obviously less than zero for $\lambda \geq 0$, i.e. the dual function is concave for $\lambda \geq 0$.

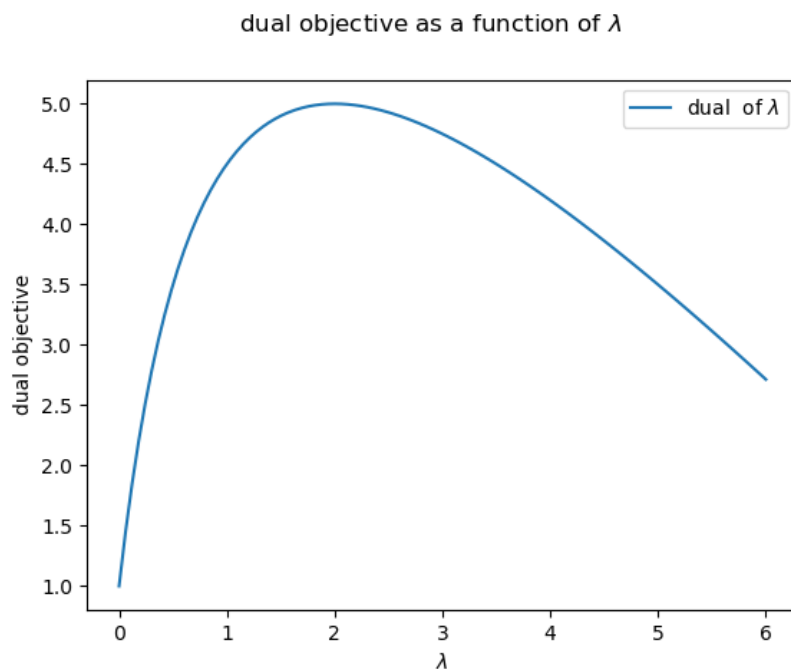


Figure 2: Dual objective for this problem.

The dual optimal value λ^* can be found solving the dual problem for the $g(\lambda)$:

$$\begin{array}{ll} \text{maximize} & g(\lambda) \\ \text{subject to} & \lambda \geq 0 \end{array}$$

or

$$\begin{array}{ll} dg(\lambda)/d\lambda = 0 \\ \text{subject to} & \lambda \geq 0 \end{array}$$

Solving the equation we found $\lambda^* = 2$; optimal value of g (i.e. $\sup_{\lambda}\{g(\lambda)\}$) $g^* = 5$. We can see that $p^* = 5 = g^*$, i. e. strong duality holds.

(d)

Solving the constraints equation $(x - 2)(x - 4) = u$ we get:
 $x_{1,2} = 3 \pm \sqrt{1 + u}$, $u \geq -1$. Then

$$p^*(u) = \begin{cases} \text{not exists,} & \text{if } u < -1 \\ u - 6\sqrt{1 + u} + 11, & \text{if } -1 \leq u \leq 8 \\ 1, & \text{if } u > 8 \end{cases} \quad (1)$$

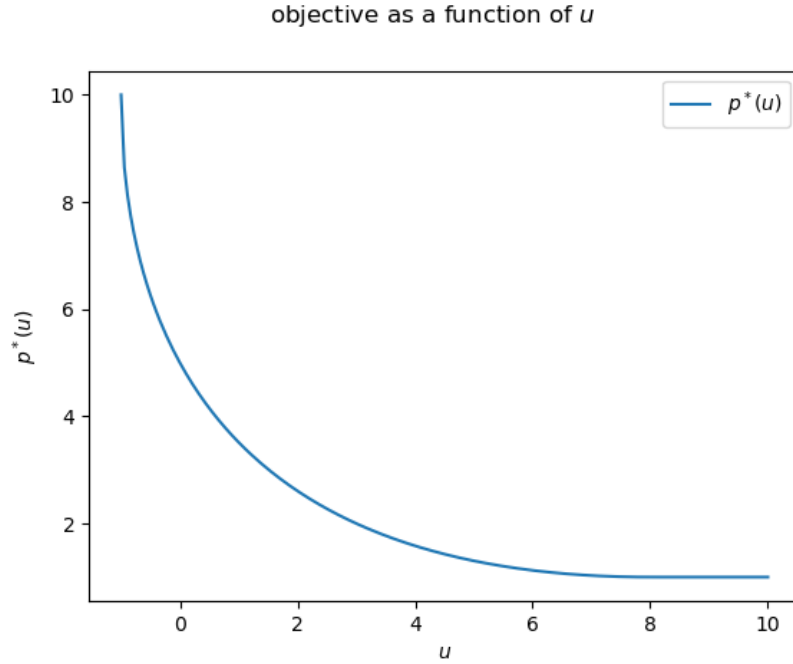


Figure 3: Graph of $p^*(u)$ for this problem.

And finally:

$$dp^*(u)/du = 1 - \frac{3}{\sqrt{1 + u}}$$

Then $dp^*(0)/du = -2 = -\lambda^*$

5.3

Problems with one inequality constraint. Express the dual problem of

$$\begin{aligned} & \text{minimize } c^T x \\ & \text{subject to } f(x) \leq 0 \end{aligned}$$

with $c \neq 0$ in terms of the conjugate f^* . Explain why the problem you give is convex. We do not assume f is convex.

Solution:

The dual problem of the task is:

$$\begin{aligned} & \text{maximize } \inf_x (c^T x + \lambda f(x)) \\ & \text{subject to } \lambda \geq 0 \end{aligned}$$

The definition of the conjugate function is:

$$f^*(y) = \sup_x (y^T x - f(x)) = - \inf_x (f(x) - y^T x)$$

i.e. the dual problem can be reformulated as:

$$\begin{aligned} & \text{maximize } F(c, \lambda) \\ & \text{subject to } \lambda \geq 0 \end{aligned}$$

where $F(c, \lambda) = -\lambda f^*(-c/\lambda)$, where f^* is the conjugate function of f , and it is always convex. $F(c, \lambda)$ is the concave function, as it is the negative perspective of the convex function f^* .

5.12

Analytic centering. Derive a dual problem for

$$\text{minimize } - \sum_{i=1}^m \log(b_i - a_i^T x)$$

with domain $\{x \mid a_i^T x \leq b_i, i = 1, \dots, m\}$. First introduce new variables y_i and equality constraints $y_i = b_i - a_i^T x$. (The solution of this problem is called the analytic center of the linear inequalities $a_i^T x \leq b_i, i = 1, \dots, m$). Analytic centers have geometric applications and play an important role in barrier methods.

Solution:

This problem is equivalent to:

$$\begin{array}{ll} \text{minimize} & -\sum_{i=1}^m \log(y_i) \\ \text{subject to} & y + Ax - b = 0 \end{array}$$

where the matrix A composed from the rows a_i^T , the i -th row of A is a_i^T . Then the Lagrangian is

$$L(x, y, \nu) = -\sum_{i=1}^m \log(y_i) + \nu^T(y + Ax - b)$$

The dual function is

$$g(\nu) = \inf_{x, y} \left(-\sum_{i=1}^m \log(y_i) + \nu^T(y + Ax - b) \right)$$

The term $\nu^T Ax$ is unbounded below as $x \rightarrow \infty$, so

$$g(\nu) = \begin{cases} \sum_{i=1}^m \log(y_i) + m - \nu^T b, & A\nu = 0, \nu_i \geq 0 \\ -\infty, & A\nu \neq 0 \end{cases}$$

and the dual problem is:

$$\begin{array}{ll} \text{maximize} & \sum_{i=1}^m \log(\nu_i) + m - \nu^T b \\ \text{subject to} & A\nu = 0 \end{array}$$

A3.3

Reformulating constraints in CVX*. Each of the following CVX code fragments describes a convex constraint on the scalar variables x , y , and z , but violates the CVX rule set, and so is invalid. Briefly explain why each fragment is invalid. Then, rewrite each one in an equivalent form that conforms to the CVX rule set. In your reformulations, you can use linear equality and inequality constraints, and inequalities constructed using CVX functions. You can also introduce additional variables, or use LMIs. Be sure to explain (briefly) why your reformulation is equivalent to the original constraint, if it is not obvious. Check your reformulations by creating a small problem that includes these constraints, and solving it using CVX. Your test problem doesn't have to be feasible; it's enough to verify that CVX processes your constraints without error. Remark. This looks like a problem about 'how to use CVX software', or 'tricks for using

CVX'. But it really checks whether you understand the various composition rules, convex analysis, and constraint reformulation rules.

- (a) $\text{norm}([x + 2y, x - y]) == 0$
- (b) $\text{square}(\text{square}(x + y)) \leq x - y$
- (c) $1/x + 1/y \leq 1; x \geq 0; y \geq 0$
- (d) $\text{norm}([\max(x,1), \max(y,2)]) \leq 3x + y$
- (e) $x*y \geq 1; x \geq 0; y \geq 0$
- (f) $(x + y)^2/\sqrt{y} \leq x - y + 5$
- (g) $x^3 + y^3 \leq 1; x \geq 0; y \geq 0$
- (h) $x + z \leq 1 + \sqrt{x*y - z^2}; x \geq 0; y \geq 0$

Solution:

- (a) $\text{norm}([x + 2y, x - y]) == 0$

This constraint is invalid because **Three types of constraints may be specified in disciplined convex programs:**

- An equality constraint, constructed using $==$, where both sides are affine.
- A less-than inequality constraint, using \leq , where the left side is convex and the right side is concave.
- A greater-than inequality constraint, using \geq , where the left side is concave and the right side is convex.

<http://web.cvxr.com/cvx/doc/dcp.html#constraints>

Expression in (a) violates the first rule - both sides must be affine. As the given norm could be zero only if $(x, y) = (0, 0)$, the workaround:

```
subject to
x == 0;
y == 0;
```

The test script:

```
cvx_begin
variable x;
variable y;
minimize(norm(x + y + 1));
subject to
x == 0;
y == 0;
cvx_end
```

```

fprintf('status:');
disp(cvx_status);

fprintf('optimal value:');
disp(cvx_optval )

fprintf('optimal x:\n');
disp(x)

fprintf('optimal y:\n');
disp(y)

```

(b) $\text{square}(\text{square}(x + y)) \leq x - y$
 this code violates the one of the composition ruleset:
<http://web.cvxr.com/cvx/doc/dcp.html#expression-rules>
If the function is neither nondecreasing or nonincreasing in an argument, that argument must be affine.

So, use instead $\text{power}(x + y, 4) \leq x - y$

The test script:

```

cvx_begin
variable x;
variable y;
minimize(norm(x + y + 1));
subject to
power(x + y, 4) <= x - y
cvx_end

fprintf('status:');
disp(cvx_status);

fprintf('optimal value:');
disp(cvx_optval )

fprintf('optimal x:\n');
disp(x)

fprintf('optimal y:\n');
disp(y)

(c)  $1/x + 1/y \leq 1;$ 

```

this expression violates the rule: **a valid call to a function in the atom library with a convex result**
<http://web.cvxr.com/cvx/doc/dcp.html#expression-rules>

Workaround: use
(c) `inv_pos(x)` or `pow_p(x,-1)` instead $1/x$

The test script:

```
cvx_begin
variable x;
variable y;
minimize(norm(x + y + 1));
subject to
inv_pos(x) + inv_pos(y) <= 1;
x >= 0;
y >= 0;
cvx_end

fprintf('status:');
disp(cvx_status);

fprintf('optimal value:');
disp(cvx_optval )

fprintf('optimal x:\n');
disp(x)

fprintf('optimal y:\n');
disp(y)

(d) norm([max(x,1), max(y,2)]) <= 3*x + y
```

The problem is that `norm()` is neither nondecreasing or nonincreasing and it can accept only affine arguments.

If the function is neither nondecreasing or nonincreasing in an argument, that argument must be affine
<http://web.cvxr.com/cvx/doc/dcp.html#constraints>)

Workaround: use
`z1 >= max(x,1); z2 >= max(y,2) norm([z1, z2]) <= 3*x + y;`

The test script:

```
cvx_begin
variable x;
variable y;
minimize(norm(x + y + 1));
subject to
z1 >= max(x,1);
```

```

z2 >= max(y,2)
norm([z1, z2]) <= 3*x + y;
cvx_end

fprintf('status:');
disp(cvx_status);

fprintf('optimal value:');
disp(cvx_optval )

fprintf('optimal x:\n');
disp(x)

fprintf('optimal y:\n');
disp(y)

(e)  $x*y \geq 1; x \geq 0; y \geq 0$ 

```

The $x*y$ function is neither convex nor concave, because its Hessian is not positive semidefinite because it has $\lambda_{1,2} = \pm 1$. But we can express this inequality as $x \geq 1/y$ or $x \geq \text{inv_pos}(y)$. The functions x and $1/y$ are both convex.

The test script:

```

cvx_begin
variable x;
variable y;
minimize(norm(x + y + 1));
subject to
x >= inv_pos(y);
x >= 0;
y >= 0;
cvx_end

fprintf('status:');
disp(cvx_status);

fprintf('optimal value:');
disp(cvx_optval )

fprintf('optimal x:\n');
disp(x)

fprintf('optimal y:\n');
disp(y)

(f)  $(x + y)^2/\sqrt{y} \leq x - y + 5$ 

```

This is not a **valid call to a function in the atom library with a convex result**. To make it a valid call we should use the `quad_over_lin((x + y), sqrt(y))` function from the atom library. I do not know why it's working in this case.

The test script:

```
cvx_begin
variable x;
variable y;
minimize(norm(x + y + 1));
subject to
quad_over_lin((x + y), sqrt(y)) <= x - y + 5;
y >= 0;
x >= y - 5;
cvx_end

fprintf('status:');
disp(cvx_status);

fprintf('optimal value:');
disp(cvx_optval )

fprintf('optimal x:\n');
disp(x)

fprintf('optimal y:\n');
disp(y)
```

(g) $x^3 + y^3 \leq 1$; $x \geq 0$; $y \geq 0$
 CVX rejects this code because the function x^3 is not convex when $x < 0$. The functions `POW_P`, `POW_POS`, or `POW_ABS` can be used instead.

The test script:

```
cvx_begin
variable x;
variable y;
minimize(norm(x + y + 1));
subject to
pow_p(x, 3) + pow_p(y, 3) <= 1;
x >= 0;
y >= 0;
cvx_end

fprintf('status:');
```

```

disp(cvx_status);

fprintf('optimal value:');
disp(cvx_optval )

fprintf('optimal x:\n');
disp(x)

fprintf('optimal y:\n');
disp(y)

```

$$(h) \ x + z \leq 1 + \sqrt{x*y - z^2}; \ x \geq 0; \ y \geq 0$$

The problem is that $x*y$ is not a concave function. But we can reformulate $\sqrt{x*y - z^2}$ as $\sqrt{x * (y - z^2 / x)}$. Then $\sqrt{}$ can be replaced with `geo_mean(x, (y - quad_over_lin(z, x)))`. This works because `geo_mean` is concave and not decreasing and can accept a concave functions as arguments, and `y - quad_over_lin(z, x)` is concave.

The test script:

```

cvx_begin
variable x;
variable y;
variable z;
minimize(norm(x + y + z + 1));
subject to
x + z <= 1 + geo_mean([x, (y - quad_over_lin(z, x))])
x >= 0;
y >= 0;
cvx_end

fprintf('status:');
disp(cvx_status);

fprintf('optimal value:');
disp(cvx_optval )

fprintf('optimal x:\n');
disp(x)

fprintf('optimal y:\n');
disp(y)

fprintf('optimal z:\n');
disp(z)

```

A3.10

3.10 Linear programming with random cost vector. We consider the linear program

$$\begin{array}{ll}\text{minimize} & c^T x \\ \text{subject to} & Ax \succ 0\end{array}$$

Here, however, the cost vector c is random, normally distributed with mean $E(c) = c_0$ and covariance $E((c - c_0)(c - c_0)^T) = \Sigma$, (A, b, x are deterministic). Thus, for a given $x \in R^n$ the cost $c^T x$ is a scalar random Gaussian variable. We can attach several different meanings to the goal ‘minimize $c^T x$ ’; we explore some of these below.