Convex Optimisation Problems

COMP4691-8691
School of Computer Science, ANU

1 Convex Functions

For a function $f : \mathbb{R}^n \longrightarrow \mathbb{R}^m$, state a condition that f needs to satisfy in order to be a convex function if f is not differentiable. State a different condition for the case where f *is* differentiable.

For the first part you could use: the line segments between any two points on graph of f, must be above or equal to f. A second option would be to state that the epigraph of f must be a convex set.

When f is differentiable, different conditions from the above are: the Hessian of f must be positive semidefinite, or the graph of f must lie on or above all its tangent planes.

2 Is a Convex Function?

Prove whether or not $f(x, y) := x^2 + xy - 2$ is convex.

Take the Hessian of the function and show it is not positive semi-definite, e.g., by using a point such as $(1, -2)$.

3 Penalty Method

Explain the penalty method, using the following all-inequality constrained problem as an example:

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f(x) \\ \text{s.t.} \quad & g(x) \leq 0 \end{aligned}$$

The Penalty Method turns a constrained optimisation problem into an unconstrained one, for a penalty parameter η_k, by applying a quadratic penalty to constraint violations:

$$\min_{x \in \mathbb{R}^n} \quad f(x) + \eta_k \sum_i^m \max(0, g_i(x))^2$$

A sequence of the above penalty problems are solved for increasing η_k, until the constraint violations are within some tolerance.

4 Dual Problem

Write out the dual problem, in terms of the Lagrangian $\mathcal{L}(x, \mu, \lambda)$ for a minimisation problem. What is the relationship between the solution of the dual and primal problem? Briefly explain the steps of the dual gradient ascent algorithm.

$$\max_{\mu \geq 0, \lambda} \inf_x \mathcal{L}(x, \mu, \lambda)$$

The dual problem will have an optimal solution that provides a lower bound on the original primal problem.

The dual gradient ascent method iterates between solving the inner level problem for the primal variables (provided the current iterate for the dual variables), and performing a gradient ascent step on the dual variables.

5 KKT

Show that the KKT conditions are satisfied for the following problem when $(x, y) = (1.5811, 0.6325)$. What are the implications of this?

$$\begin{aligned} \min_{x,y} \ & (y+2)^2 \\ \text{s.t. } & y \geq \frac{1}{x} \\ & 2x^2 \leq 5 \\ & x \geq 1 \end{aligned}$$

Using μ_1, μ_2, μ_3 for the inequality constraints above respectively. The stationarity condition consists of the following two equations:

$$\begin{aligned} -\mu_1 \frac{1}{x^2} + 4\mu_2 x - \mu_3 &= 0 \\ 2(y+2) - \mu_1 &= 0 \end{aligned}$$

The constraints are:

$$\begin{aligned} \frac{1}{1.5811} - 0.6325 &= 0.000 \\ 2 \cdot 1.5811^2 - 5 &= 0.000 \\ 1 - 1.5811 &= -0.5811 \end{aligned}$$

All constraints are satisfied (**primal feasibility**). The last constraint is not active, so due to the complementary slackness condition, its dual variable must be zero $\mu_3 = 0$ (**complementary slackness**). Plugging in the candidate point to the complementary slackness terms we get the following expressions for the other two duals:

$$\begin{aligned} \mu_2 &= 2(y+2)\frac{1}{4x^3} = 0.333 \\ \mu_1 &= 2(y+2) = 5.265 \end{aligned}$$

All duals are positive (**dual feasibility**) and the stationarity condition is satisfied with these dual values (**stationarity**).

The problem is convex and all functions are smooth over the domain of interest, so the KKT conditions are sufficient for a global optimal solution.

6 Convex Relaxation

Take a convex relaxation of the following set of constraints (need not be linear):

$$y = \cos(\theta)$$
$$\theta \in [-\frac{\pi}{2}, \frac{\pi}{2}]$$

$$y \leq \cos(\theta)$$
$$y \geq 0$$
$$\theta \in [-\frac{\pi}{2}, \frac{\pi}{2}]$$

7 Method Order

Discuss the tradeoffs between first and second order methods.

Second order methods can often achieve higher rates of convergence (super-linear convergence rates) over first order methods. However, this comes at the cost of more computation per iteration, e.g., in calculating Hessians and solving a large linear system of equations. First order methods might be necessary for applications where the problem is very large, where the Hessian may struggle to fit into memory.

8 Dual Gradient Ascent

Dual gradient ascent is an algorithm for solving strictly convex constrained optimisation problems. In this question you will implement the approach, and use it to solve several problems.

The `problems.py` file contains two constrained optimisation problems hard-coded into classes `Problem1` and `Problem2`. These classes offer functions to evaluate the objective, the gradient of the objective, the constraints and the constraint jacobian. This file also contains a more general class to represent

quadratically constrained quadratic programs (QCQP). It can load a specific problem instance from file.

Your code should go in the file `dual_ascent.py`. This can be called as a script. If you provide it with a JSON file it will solve the corresponding QCQP, otherwise it will solve the two hardcoded `Problem1` and `Problem2`.

8.1 Gradient Descent

As a subcomponent, we need to be able to first solve gradient descent. Implement the gradient descent algorithm in the `gradient_descent` function, of the `dual_ascent.py` file.

Solve unconstrained versions of `Problem1` and `Problem2` (just ignore the constraints) using your gradient descent algorithm. Report the number of iterations required and the solution.

Hint: You'll likely want to make use of one of the rule-based step sizes, e.g., one of those by Barzilai and Borwein from the lectures.

Hint: Implement a stopping criteria that checks whether the gradient is within a provided tolerance of being zero.

Instance	Objective	Primals	GD Iter
Problem1	0	-2	3
Problem2	0.845,	-0.278, 0.273	12

8.2 Dual Gradient Ascent

Next implement the dual ascent algorithm in `dual_ascent`, making use of `gradient_descent`.

Solve the **constrained** versions of `Problem1` and `Problem2` using your algorithm. Report the number of iterations required and the solution.

Hint: Implement a stopping criteria based on satisfaction of the KKT conditions to within a tolerance.

Instance	Objective	Primals	Duals	DA Iter	GD Iter
Problem1	9.0	1	1.2	28	84
Problem2	174.9	5, 6.38	1.43, 138.3	639	3302

8.3 More General Quadratic Problems

Report the results you get by running your dual ascent algorithm on instances `qcqp-{1,2,3,4}.json`. Based on the dual variables you obtain, explain which constraints are active for each instance. You will likely struggle to solve the instance `qcqp-3.json`. If so explain why.

Instance	Objective	Primals	Duals	DA Iter	GD Iter
qcqp-1	0.2055	0.6411, 0, 0	0.1471	10	28
qcqp-2	-0.5	-1, 0, 0	0	1	3
qcqp-3	Not solved				
qcqp-4	-2.6	-0.15, 0.63, 0.91, 0.18	0, 0.56	21	359

Any non-zero dual variables indicate an active constraint. For the first problem the constraint is active, for the second problem the constraint is not active, and for the forth problem the second constraint is active.

Instance 3 is convex, but not strictly convex, which causes trouble for our algorithms in two respects. Firstly, the rule-based step size calculation will have to fall back to a default value in parts of the search space, which can lead to slow convergence. Secondly, the dual function can be unbounded below for particular dual values, which will make the solve process unstable and will likely prevent dual ascent from converging. In our example, with the default starting values, the primal x_2 and x_3 variables jumped up to very large values, likely as a result of an unbounded subproblem that terminated after 1000 iterations. From then onwards the algorithm struggled to bring them back down to sensible values.