UNIVERSITÄT BERN

Applied Optimization
Exercise 2 - Convex Functions

Heng Liu   Nicolas Gallego

Handout date: 03.10.2019

Submission deadline: 09.10.2019, 23:59 h

Hand-in instructions:

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

- **Only** the files you changed (headers and source). It is up to you to make sure that all files that you have changed are in the zip.
- 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.

Convex Functions (6 pts)

First-order condition (1 pt)

Prove the first-order condition: A differentiable function f is convex if and only if $\mathbf{dom} f$ is convex and

$$f(y) \geq f(x) + \nabla f(x)^T (y - x)$$

holds for all $x, y \in \mathbf{dom} f$.

Hint: First consider $f : \mathbb{R} \to \mathbb{R}$ and the definition of the derivative. Then for $f : \mathbb{R}^n \to \mathbb{R}$ consider f constrained to a line.

Second-Order Condition (1 pt)

Show that a function f, twice differentiable, is convex if and only if $\mathbf{dom} f$ is convex and its Hessian is positive semidefinite for all $x \in \mathbf{dom} f$

$$\nabla^2 f \succeq 0$$

Hint: Consider also first $f : \mathbb{R} \to \mathbb{R}$ and you can also use the first order condition.

Log-sum-exp (2 pt)

Show that the function

$$f(\mathbf{x}) = \log(e^{x_1} + \ldots + e^{x_n})$$

is convex on $\mathbb{R}^n$.

Hint: The proof is outlined in the Boyd's book p. 74. Develop by it yourself and provide the intermediate steps.

Bonus (2 pts)

Geometric mean

Show that the geometric mean

$$f(\mathbf{x}) = (\Pi_{i=1}^{n} x_i)^{1/n}$$

is concave on $\mathbb{R}^n_{++}$.

Programming Exercise: Mass Spring System (6 pts)

Consider a system of springs connecting nodes of an m by n grid with connectivity as illustrated in Figure **??**. The coordinate $\mathbf{x}$ of a node indexed a at grid (i, j), where $a \in (0, ..., (m+1)*(n+1)-1)$, $i \in (0, ..., m)$and $j \in (0, ..., n)$, is not fixed in $\mathbb{R}^2$. Each edge of the grid has an elastic constant k and each node is subject to an elastic force proportional to the distance between the nodes. Let $\mathbf{x_a}$ and $\mathbf{x_b}$ denote the positions of nodes a and b respectively connected by an edge with elastic constant $k_{a,b}$. Compute the total potential energy stored in the system for a given set of positions under the following modeling assumptions: (1) ideal springs without length and (2) springs with length. Denote with $||\mathbf{x_a} - \mathbf{x_b}||$ the euclidean distance between $\mathbf{x_a}$ and $\mathbf{x_b}$.

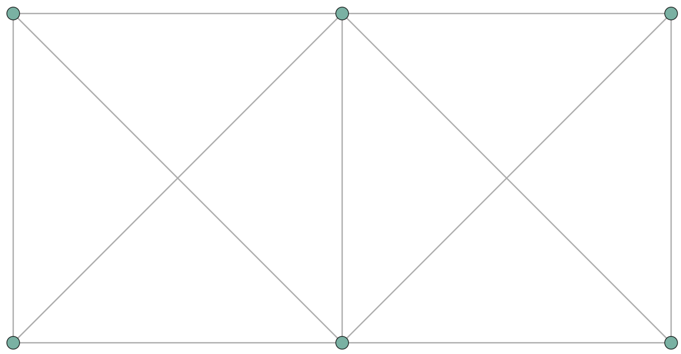

Figure 1: Example: 2×1 grid

1. Springs without length In this case magnitude of the force between this nodes will be given by:

$$F_{a,b} = k_{a,b}||\mathbf{x_a} - \mathbf{x_b}||,$$

and the potential energy between nodes

$$E_{a,b} = \frac{1}{2}k_{a,b}||\mathbf{x_a} - \mathbf{x_b}||^2,$$

Thus the total energy in the system is the sum of the energies of the individual edges $e(a,b)$

$$E = \sum_{e(a,b)} E_{a,b}$$

2. Springs with length Now let the rest length of the spring be $l_{i,j}$. The magnitude of the force will be in this case

$$\hat{F}_{a,b} = k_{a,b}(||\mathbf{x_a} - \mathbf{x_b}|| - \mathbf{l_{a,b}}),$$

and the potential energy between nodes

$$\hat{E}_{a,b} = \frac{1}{2}k_{a,b}(||\mathbf{x_a} - \mathbf{x_b}||^2 - \mathbf{l_{a,b}^2})^2,$$

Thus the total energy in the system is the sum of the energies of the individual edges $e(i,j)$

$$\hat{E} = \sum_{e(a,b)} \hat{E}_{a,b}$$

In this exercise, you are request to check if these two energy functions E and $\hat{E}$ are convex with the second-order condition. The first step is to implement the `setup_spring_graph()` function in `MassSpringSystem.cc` to construct the spring graph. You can make use of the data structure provided in `SpringGraph.hh`. Make sure the spring graph is correctly setup.
The system can be viewed as a combination of elements, with each spring being an element. In `SpringElement2D.hh` and `SpringElement2DWithLength.hh`, with the energy functions $E_{a,b}$ and $\hat{E_{a,b}}$ given above, the functions `f(...)`, `grad_f(...)` and `hess_f(...)` should be implemented accordingly. Then the local data of each element is assembled into the MassSpringProblem. There are two implementations of this problem, one with dense hessian matrix and the other with sparse hessian matrix. You should fill out the corresponding functions in the two files `MassSpringProblem2D.hh` and `MassSpringProblem2DSparse.hh`. From what we've learned so far, one natural way to store the hessian matrix is to use eigen dense matrix. With the `eigenvalues()` function shipped with dense matrix type, you can easily check if the hessian matrix is positive semi-definite by examine the smallest eigenvalue. However, since the hessian matrix of the energy function is sparse, a smarter way is to employ the eigen sparse matrix type. Together with the `spectra` library, the eigenvalue calculation would be much more efficient.
In the end, you should be able to implement the `is_convex()` function in `MassSpringSystem.cc` with the MassSpringProblem classes. Regarding the elastic constant k and the length l, for simplicity all k is set to 1 and the diagonal edge length l is set to $\sqrt{2}$ and the rest is 1.