Convex Optimization for Machine Learning and Computer Vision

Lecture: Dr. Tao Wu — Computer Vision Group
Exercises: Zhenzhang Ye — Institut für Informatik
Winter Semester 2019/20 — Technische Universität München

Weekly Exercises 2

Room: 02.09.023

Wednesday, 06.11.2019, 12:15-14:00

Submission deadline: Monday, 04.11.2019, 16:15, Room 02.09.023

Theory: Convex Functions (10+8 Points)

Exercise 1 (4 Points)**.** Let $J : \mathbb{E} \to \bar{\mathbb{R}}$ be proper. Prove the equivalence of the following statements:

- J is lower semi-continuous (l.s.c).
- The epigraph of J is closed.

Exercise 2 (4 Points)**.** Suppose $J : \mathbb{E} \to \mathbb{R}$ is convex with $\text{dom}J = \mathbb{R}^n$, and bouned above on $\mathbb{R}^n$. Show that J is a constant function.

Exercise 3 (6 Points)**.** Show that the following functions $J : \mathbb{E} \to \overline{\mathbb{R}}$ are convex:

- $J(u) = \|u\|$, for any norm $\|\cdot\|$ over a normed vector space.
- $J(u) = F(Ku)$, for convex $F : \mathbb{E} \to \overline{\mathbb{R}}$ and linear $K : \mathbb{R}^m \to \mathbb{E}$.
- $J(u) = \max\{J_1(u), J_2(u)\}$, where J_1 and J_2 are convex functions with $\mathbb{E} \to \overline{\mathbb{R}}$.

Exercise 4 (4 Points)**.** Let $U \subset \mathbb{E}$ open and convex and let $J : U \to \mathbb{R}$ be twice continuously differentiable. Prove the equivalence of the following statements:

- J is convex.
- For all $u \in U$ the Hessian $\nabla^2 J(u)$ is positive semidefinite ($\forall\, v \in \mathbb{E} : v^\top \nabla^2 J(u) v \geq 0$).

Hints: You can use that for $u, v \in U$ it holds that J is convex iff

$$(v-u)^\top \nabla J(u) \leq J(v) - J(u).$$

Further recall that there are two variants of the Taylor expansion:

$$J(u+td) = J(u) + td^\top \nabla J(u) + \frac{t^2}{2} d^\top \nabla^2 J(u) d + o(t^2)$$

with $\lim_{t\to 0} \frac{o(t^2)}{t^2} = 0$ and

$$J(u+d) = J(u) + d^\top \nabla J(u) + \frac{1}{2} d^\top \nabla^2 J(u+td) d$$

for appropriate $t \in (0,1)$.

Programming: Inpainting(Due date: 11.11) (12 Points)

Exercise 5 (12 Points)**.** Write a program in MATLAB (or Python) that solves the inpainting problem for the vegetable image:

$$\min_{u \in \mathbb{R}^{n \times m}} \sum_{i,j} (u_{i,j} - u_{i-1,j})^2 + (u_{i,j} - u_{i,j-1})^2 \quad \text{s.t. } u_{i,j} = f_{i,j} \ \forall (i,j) \in I,$$

with index set I of pixels to keep. Those can be identified as the white pixels of the mask image.
Hint: The constrained optimization problem can be reformulated so that it becomes unconstrained: Rewrite the objective as a least squares problem in terms of the unknown intensities $u_{i,j}$, $(i,j) \notin I$ using sparse linear operators: Find linear operators X, Y s.t. u can be decomposed as

$$u = X\tilde{u} + Yf$$

where $\tilde{u}$ contains only the unknown intensities. Optimize for $\tilde{u}$ instead of u. You may use MATALBs `mldivide` (for Python, check out e.g. `scipy.sparse.linalg.{spsolve, lsqr}`).