Convex Optimization for Machine Learning and Computer Vision

Lecture: T. Wu
Exercises: E. Laude, T. Möllenhoff
Summer Semester 2017

Computer Vision Group
Institut für Informatik
Technische Universität München

Weekly Exercises 2

Room: 02.09.023

Monday, 15.05.2017, 12:15-14:00

Submission deadline: Wednesday, 10.05.2017, 16:15, Room 02.09.023

Convex sets and functions (9 Points + 4 Bonus)

Exercise 1 (4 Points)**.** Let $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ be proper. Prove the equivalence of the following statements:

- f is convex.
- $\operatorname{epi}(f) := \left\{ \begin{pmatrix} x \\ y \end{pmatrix} \in \mathbb{R}^{n+1} : f(x) \leq y \right\}$ is convex.

Exercise 2 (3 Points)**.** Show that the following functions $f : \mathbb{R}^n \to \overline{\mathbb{R}}$ are convex:

- $f(x) = \|x\|$, for any norm $\|\cdot\|$.
- $f(x) = g(Ax)$, for convex $g : \mathbb{R}^m \to \overline{\mathbb{R}}$ and linear $A : \mathbb{R}^n \to \mathbb{R}^m$.
- The perspective of a convex function $g : \mathbb{R}^{n-1} \to \overline{\mathbb{R}}$ given as

$$f(x,t) := \begin{cases} t\, g\left(\frac{x}{t}\right), & \text{if } t > 0 \text{ and } \frac{x}{t} \in \operatorname{dom}(g), \\ +\infty, & \text{otherwise.} \end{cases}$$

Exercise 3 (2 Points)**.** Let $\mathbb{E}$ be an Euclidean space. Show that the following two statements are equivalent:

- $f : \mathbb{E} \to \overline{\mathbb{R}}$ is convex,
- $f\left(\sum_{i=1}^n \alpha_i x_i\right) \leq \sum_{i=1}^n \alpha_i f(x_i)$, for $x_i \in \mathbb{E}$, $\alpha_i \in [0,1]$, $\sum_{i=1}^n \alpha_i = 1$, $n \geq 1$.

Exercise 4 (4 points)**.** Prove the following statement using induction over m: Let $K_1, \ldots, K_m \subset \mathbb{R}^n$, $m \geq n+1$, be convex, such that for all $\mathcal{I} \subset \{1, \ldots, m\}$ with $|\mathcal{I}| = n+1$ it holds that $\bigcap_{i \in \mathcal{I}} K_i \neq \emptyset$. Then $\bigcap_{i=1}^m K_i \neq \emptyset$.

Hint: Use exercise 4 from the first exercise sheet.

Image Cartooning (12 Points)

Exercise 5 (12 Points)**.** In this exercise your task is to compute a piecewise constant, cartoonish looking approximation of the input image (consisting of n pixels). This can be done as follows: We begin selecting k different colors $\{c_1, c_2, \dots c_k\} \subset \mathbb{R}^3$ that are most present in the image, for example $c_1 =$ red, $c_2 =$ green, $c_3 =$ blue and $c_4 =$ yellow. We then segment the image into k disjoint regions, so that the overall boundary length is short and at the same time the pixels in the i-th region are close to the i-th color.

As explained in the lecture (cf. chapter 0), one can solve the following optimization problem

$$\min_{u \in \mathbb{R}^{k \times n}} \langle u, f \rangle + \sum_{j=1}^{n} \delta\{u(:,j) \in \Delta^k\} + \alpha \sum_{i=1}^{k} \|Du(i,:)\|_1 \tag{1}$$

where $f \in \mathbb{R}^{k \times n}$, f_{ij} is given as the Euclidean distance of pixel j to color i, $u(i,:) \in \mathbb{R}^n$ is the i-th row of u and $u(:,j) \in \mathbb{R}^k$ is the j-th column. The matrix $D : \mathbb{R}^n \to \mathbb{R}^{2n}$ computes the gradient using finite differences (see the previous sheet).

Let $\tilde{u} \in \mathbb{R}^{k \times n}$ be a minimizer of problem (1). The cartoon image $\bar{u} \in \mathbb{R}^{3 \times n}$ is given as

$$\bar{u}(:,j) := c_m, \quad \text{where} \quad m := \arg\max_{1 \le i \le k} \tilde{u}(i,j), \quad \forall 1 \le j \le n. \tag{2}$$

In this exercise, you will use projected gradient descent to solve the above problem. This algorithm performs (A) a gradient descent step on the differentiable part of the energy, followed by a projection (B) onto the simplex Δ^k.

$$\begin{aligned} &\text{(A)} \quad u^{t+\frac{1}{2}} = u^t - \tau \nabla E(u^t), \\ &\text{(B)} \quad u(:,j)^{t+1} = \arg\min_{u \in \Delta^k} \|u - u^{t+\frac{1}{2}}(:,j)\|, \ \forall 1 \le j \le n. \end{aligned} \tag{3}$$

Since the norm $\|\cdot\|_1$ is not differentiable at 0, we use a smooth approximation. The function E is given as:

$$E(u) = \langle u, f \rangle + \alpha \sum_{i=1}^{k} \|Du(i,:)\|_\varepsilon,$$

where $\|x\|_\varepsilon = \sum_i \sqrt{x_i^2 + \varepsilon}$. Proceed as follows:

1. Load a color input image and compute the colors $c_i \in \mathbb{R}^3$ using k-means clustering (MATLAB: use the `kmeans` command). From that, construct the matrix $f \in \mathbb{R}^{k \times n}$ as described above.

2. Compute the gradient ∇E of the function $E : \mathbb{R}^{k \times n} \to \mathbb{R}$. *Hint:* you can do this separately for each component $1 \le i \le k$.

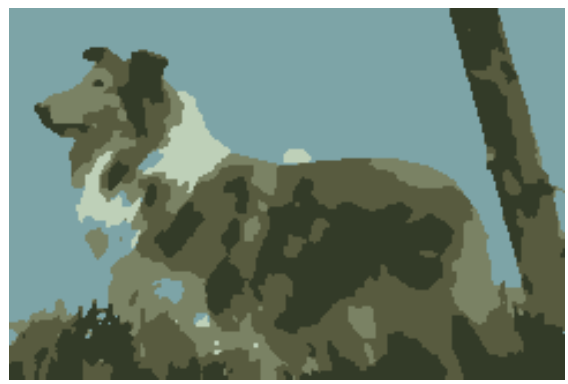
3. Implement the projected gradient descent method (3). For the projection onto the simplex (B), use the supplied helper file `projSimplex.m`.

4. Run the algorithm to compute an approximate minimizer of (1). Terminate the iteration once $\frac{1}{n \cdot k} \left\| u^{t+1} - u^t \right\|_1 < \texttt{tol}$ for some tolerance $\texttt{tol} > 0$.

5. Compute $\bar{u} \in \mathbb{R}^{3 \times n}$ as in (2) and visualize it as a color image. Experiment with different step sizes τ, parameters $\varepsilon, \alpha, k, \texttt{tol}$ and initialisations $u^0 \in \mathbb{R}^{k \times n}$. What do you observe?

The resulting $\bar{u} \in \mathbb{R}^{3 \times n}$ for $k = 5$, $\varepsilon = 0.1$, $\tau = 0.25$, $\alpha = 0.1$ and $\texttt{tol} = 10^{-6}$ should look like the following:

input image



$\bar{u} \in \mathbb{R}^{3 \times n}$