Convex Analysis and Optimization

http://www.mia.uni-saarland.de/Teaching/cao15.shtml

Dr. Peter Ochs

— Theoretical Assignment 3 —

Submission deadline:
26/11/2015
end of the lecture

Exercise 1: (5 points)
Prove or disprove convexity of the following functions.

- $f: X \to \overline{\mathbb{R}}$, $x \mapsto \langle a, x\rangle + b$ with

  (a) $X = \mathbb{R}^N$;

  (b) $X = \{0, 1\}^N$;

  (c) $X = \mathbb{Z}$ (the set of integers).

(d) $f: \mathbb{R} \times \mathbb{R} \to \overline{\mathbb{R}}$, $(x_1, x_2) \mapsto (x_1 x_2)^2$.

Prove the convexity of the following functions:

(e) Let $g_x: \mathbb{R}^N \to \overline{\mathbb{R}}$ be given by $g_x(y) = \langle x, y\rangle - \|x\|_2$ for each $x \in \mathbb{R}^N$. Show that g_x is convex for any x.

(f) Show that the function $G(y) := \sup_{x\in\mathbb{R}^N} g_x(y)$ is a convex function and provide a description of G without the "sup".

(g) Prove Theorem 4.6 from the lecture:
If f_1 and f_2 are proper convex functions on $\mathbb{R}^N$ and $\lambda_1, \lambda_2 \geq 0$, then $\lambda_1 f_1 + \lambda_2 f_2$ is convex.

Exercise 2: (3 points)
Let $C \subset \mathbb{R}^N$ be an open, convex set and let $f: \mathbb{R}^N \to \overline{\mathbb{R}}$ be a proper convex function that is continuously differentiable on $C \subset \operatorname{dom} f$. Moreover f satisfies for some $m > 0$ the inequality

$$f(x) \geq f(\bar{x}) + \langle \nabla f(\bar{x}), x - \bar{x}\rangle + \frac{m}{2}\|x - \bar{x}\|_2^2$$

for all $x, \bar{x} \in C$.

(a) Show that f satisfies

$$\langle \nabla f(x) - \nabla f(\bar{x}), x - \bar{x}\rangle \geq m\|x - \bar{x}\|_2^2 .$$

for all $x, \bar{x} \in C$.

(b) Show that the function $g(x) := f(x) - \frac{m}{2}\|x\|_2^2 + \delta_C(x)$ is a convex function.

Exercise 3: (2 points)
Show that for $\lambda \in [0, 1]$ and $x, y \in \mathbb{R}$, we have

$$x^\lambda y^{1-\lambda} \leq \lambda x + (1 - \lambda) y ,$$

with equality if and only if $x = y$. (*Hint: Show and use the convexity of* $-\ln(x)$ *on a suitable set*).

— Programming Assignment 3 —

Submission deadline:
26/11/2015
end of the lecture

Submission of Programming Results:

- Compress your filled files (zip or tar.gz) e.g. `ex03_NAME1_NAME2_NAME3.tar.gz`
- Send the compressed file to `ochs@mia.uni-saarland.de` with subject: `[CAO15] Submission ex03`

- Download `ex03.tar.gz` from `www.mia.uni-saarland.de/Teaching/cao15.shtml` and extract the files, e.g., `tar -xzvf ex03.tar.gz` .

Exercise 4: (10 points)
Consider the function $E\colon \mathbb{R}^N \to \mathbb{R}$ given by (cf. exercise sheet 01, Ex2)

$$E(u) = \frac{\lambda}{2}\|u - f\|^2 + \frac{1}{2}\|D^h u\|^2 ,$$

where $f \in \mathbb{R}^N$ and $\lambda > 0$ are given, and $D^h \in \mathbb{R}^{N\times N}$ is the matrix implementing forward differences for some $h > 0$

$$(D^h u)_i := \begin{cases} \dfrac{u_{i+1} - u_i}{h}, & \text{if } i \in \{1, \dots, N-1\}\,; \\ 0, & \text{if } i = N\,. \end{cases}$$

A minimizer of this objective function can be used to denoise a signal f, which has been deteriorated by additive Gaussian noise such that $f = g + n$, where $g \in \mathbb{R}^N$ is the clean signal and $n \in \mathbb{R}^N$ is the noise. The goal of this exercise is to implement and to compare several methods for minimizing this objective function. (*In the context of signal processing, the objective function is often called “energy function”.*)
The file `ex03.m` loads a noisy signal f, the optimal solution `x_opt`, and the optimal value `E_opt`. The signal f is equidistantly sampled in the interval $[0, \pi]$ with sample distance $h = 0.01$.

(a) Explain the role of the parameter $\lambda > 0$.

From now on, we fix $\lambda = 180$, `maxiter=200`, and use the Lipschitz constant $L = \lambda + \frac{4}{h^2}$ of ∇E.

(b) Implement the so-called Heavy-ball method in `hb.m` and the conjugate gradient method in `cg.m`. (*See last page*.) Make sure that the following quantities are taped (in the way of the previous exercises) and returned to `ex03.m`:

   – The distance of the current iterate (`x_np1`) to the optimum in `seq_err`.
   – The difference of the current function value (`E_np1=E(x_np1)`) to the optimal value in `seq_E`.

Note that the gradient descent method is a special instance of the Heavy-ball method by choosing $\beta = 0$.

(c) In the file `ex03.m`, fill the part that calls the Gradient Descent method, the Heavy-ball method, the optimal Heavy-ball method (use $m = \lambda$) and the conjugate gradient method as follows: (4 instances)

- Find the step size parameter α for the gradient descent method that yields the best performance (within the range where convergence is guaranteed).
- Find the best parameter β for the Heavy-ball method when α is fixed to $1/L$.
- Call the Heavy-ball method with optimal parameters.
- Call the Conjugate gradient method.

Now, we want to analyze the performance of these 4 methods.

(d) Analyze the convergence of the errors:

- Visualize the convergence of the errors `errs` for all 4 method in a single `loglog`-plot.
- Add the convergence rates $\mathcal{O}(\frac{1}{n^p})$, $\mathcal{O}(\omega^n)$ to the plot, where n is the number of iterations and $\omega \in (0, 1)$, $p > 0$. Use the freedom of the $\mathcal{O}$-notation to assure that the convergence rate has the same value as the errors at the very first iterate $n = 1$.
- Experimentally find the asymptotic convergence rates for all 4 methods by selecting a suitable p or ω for each of them. Report these values.

(e) Analyze the convergence of the function values in the same way as the errors by repeating the three steps in (d).

Algorithm 1: Heavy-ball method

- **Optimization problem**: ($A \in \mathbb{R}^{N\times N}$ symmetric and positive definite, $b \in \mathbb{R}^N$)

$$\min_{x\in\mathbb{R}^N} f(x)\,, \quad f(x) = \frac{1}{2}\langle x, Ax\rangle - \langle b, x\rangle$$

- **Step size parameter**: $\beta \in [0,1)$ and $\alpha \in \frac{2(1+\beta)}{L}$
  Optimal choice: ($m > 0$ is a lower bound on the eigenvalues of A)

$$\beta = \left(\frac{(\sqrt{L}-\sqrt{m})}{(\sqrt{L}+\sqrt{m})}\right)^2\,, \qquad \alpha = \frac{4}{(\sqrt{L}+\sqrt{m})^2}$$

- **Iterations** ($n \geq 0$): Update ($x^{(-1)} = x^{(0)} \in \mathbb{R}^N$)

$$x^{(n+1)} = x^{(n)} - \alpha\nabla f(x^{(n)}) + \beta(x^{(n)} - x^{(n-1)})$$

Algorithm 2: Conjugate Gradient method

- **Solve the system of equations**: ($A \in \mathbb{R}^{N\times N}$ symmetric, positive definite, $b \in \mathbb{R}^N$)

$$Ax = b$$

- **Initialization**: $r^{(0)} = p^{(0)} = b - Ax \qquad r^{(0)}, p^{(0)} \in \mathbb{R}^N$
- **Iterations** ($n \geq 0$): Update ($x^{(0)} \in \mathbb{R}^N$)

$$\begin{aligned}
\alpha^{(n)} &= \frac{\langle r^{(n)}, r^{(n)}\rangle}{\langle p^{(n)}, Ap^{(n)}\rangle}\\
x^{(n+1)} &= x^{(n)} + \alpha^{(n)}p^{(n)}\\
r^{(n+1)} &= r^{(n)} - \alpha^{(n)}Ap^{(n)}\\
\beta^{(n)} &= \frac{\langle r^{(n+1)}, r^{(n+1)}\rangle}{\langle r^{(n)}, r^{(n)}\rangle}\\
p^{(n+1)} &= r^{(n+1)} + \beta^{(n)}p^{(n)}
\end{aligned}$$