Optimization and Approximation
Final exam

15/01/2021

Part I

Exercise 1 - Newton's method (*)

1. Assume Newton's method is applied to the following problem:

$$\min_x f(x)$$

with

$$f(x) = \begin{cases} (x-4)^3 & \text{if } x \geq 2 \\ x^2 - 12 & \text{if } x < 2 \end{cases}$$

The values in Table 1 are obtained by running the method (without line-search) starting from $x_0 = 10$. How can you explain the behaviour of the method in the last iterations?

2. Is it possible to use Newton's method to solve a least-squares problem? If yes, when would it be advantageous over Gauss-Newton or Levenberg-Marquartd methods?

3. Assume the Newton's method without linesearch is applied to the function $f(x) = x^4 - x^3 - 2x^2 + 3x - 2$ starting from $x_0 = 5$ and that the generated sequence converges to $x^* = -1$. Can we deduce from this that the method is globally convergent?

Exercise 2 - Convex constrained problems (**) Consider the following constrained optimization problem:

$$\min_{\mathbf{x}} f(\mathbf{x}) \tag{1a}$$

$$\text{subject to} \tag{1b}$$

$$g_i(\mathbf{x}) \leq 0 \text{ for } i = 1, \ldots, m, \tag{1c}$$

$$h_i(\mathbf{x}) = 0 \text{ for } i = 1, \ldots, p. \tag{1d}$$

k	x_k	$f(x_k)$	$f'(x_k)$
0	10	216	108
1	9.667	181.963	96.333
2	9.314	150.037	84.707
3	8.937	120.359	73.1
4	8.532	93.099	61.624
5	8.091	68.467	50.208
6	7.602	46.738	38.925
7	7.047	28.285	27.850
8	6.390	13.660	17.143
9	5.554	3.751	7.243
10	4.267	0.019	0.213
11	-3.234	-1.541	-6.468
12	-2.925	-3.446	-5.850
13	-2.583	-5.329	-5.166
14	-2.196	-7.179	-4.391
15	-1.740	-8.972	-3.480
16	-1.166	-10.641	-2.3
17	-0.308	-11.905	-0.615
18	2.942	-1.183	3.356
19	4.833	0.578	2.082
20	2.432	-3.851	7.371
21	3.708	-0.025	0.255
22	10.567	283.269	129.396
23	10.263	245.663	117.674
24	9.944	209.969	105.980
25	9.607	176.288	94.320

Table 1: Newton method applied to f starting from $x_0 = 10$.

Assume f and g_i for $i = 1, \dots, m$ are continuously differentiable convex functions and h_i for $i = 1, \dots, p$ are affine functions ($h_i(\mathbf{x}) = \mathbf{a}_i^T \mathbf{x} + b_i$ for some $\mathbf{a}_i \in \mathbb{R}^n$ and $b_i \in \mathbb{R}$). Show that the KKT conditions are sufficient for optimality for this problem, i.e. prove that if $\mathbf{x}^*$ is a KKT point, then it is a global minimum.

Hint: consider the following problem:

$$\min_{\mathbf{x}} \mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\mu})$$

with $\mathcal{L}$ the Lagrangian function.

Exercise 3 - Gradient method (**) A function $f : \mathbb{R}^n \to \mathbb{R}$ is said to be

1. *L–smooth* if ∇f is Lipschitz continuous with Lipschitz constant L:

$$\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\| \leq L \|\mathbf{x} - \mathbf{y}\|$$

for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$.

2. *μ–strongly convex* if

$$f(\mathbf{y}) \geq f(\mathbf{x}) - \nabla f(\mathbf{x})^T(\mathbf{x}-\mathbf{y}) + \frac{\mu}{2}\|\mathbf{x}-\mathbf{y}\|^2 \tag{2}$$

   $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$.

Consider the following results.

Lemma 1. *Let $f : \mathbb{R}^n \to \mathbb{R}$ a twice differentiable function. If f is L-smooth, then the two following relations hold for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$:*

$$f(\mathbf{x}) \leq f(\mathbf{y}) + \nabla f(\mathbf{y})^T(\mathbf{x}-\mathbf{y}) + \frac{L}{2}\|\mathbf{x}-\mathbf{y}\|^2, \tag{3}$$

$$f(\mathbf{x}^*) - f(\mathbf{x}) \leq -\frac{1}{2L}\|\nabla f(\mathbf{x})\|^2. \tag{4}$$

Theorem 1. *Let $f : \mathbb{R}^n \to \mathbb{R}$ a twice differentiable function, L-smooth and μ-strongly convex. Let $\mathbf{x}^*$ be the global minimum of f. Given $\mathbf{x}_0 \in \mathbb{R}^n$ and $\frac{1}{L} \geq \alpha > 0$, the iterates*

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha\nabla f(\mathbf{x}_k) \tag{5}$$

converge according to

$$\|\mathbf{x}_{k+1} - \mathbf{x}^*\|^2 \leq (1-\alpha\mu)^{k+1}\|\mathbf{x}_0 - \mathbf{x}^*\|^2.$$

Prove convergence of the gradient method for strongly convex smooth functions:

1. Prove (3) in the lemma. Hint: use the Taylor expansion of f in integral form.
2. Prove (4) in the lemma. Hint: use (3).
3. Use the lemma to prove the theorem.

Part II

Exercise 4 - Linear problems (*) State whether each of the following statements is true or false. Motivate your answer.

1. Consider a linear program:

$$\begin{aligned} \max \quad & \mathbf{c}^T\mathbf{x} \\ \text{subject to} \quad & A\mathbf{x} \leq \mathbf{b} \\ & \mathbf{x} \geq \mathbf{0}. \end{aligned}$$

   If matrix A and vectors $\mathbf{b}$, and $\mathbf{c}$, have only integer-valued entries, then there exists an optimal solution $\mathbf{x}^*$ with only integer-valued entries.

2. It possible to remain on the same vertex in two consecutive steps of the simplex method.

3. The solution of the relaxation of an integer problem is always feasible for the original integer problem.

4. If a linear program has an unbounded feasible region then it has no optimal solution.

5. Any feasible LP in real numbers with a bounded feasible region has either one or an infinite number of optimal solutions.

6. If $\mathbf{x}^*, \mathbf{y}^*$ and z^* are respectively optimal solutions of the primal, dual and the optimal value of the LP in point 1, then $z^* = (\mathbf{y}^*)^T A\mathbf{x}^*$.

Exercise 5 - Halfspaces (****)

1. Consider the following LP:

$$\begin{aligned} \min \mathbf{b}^T\mathbf{y} \\ \text{subject to} \\ A^T\mathbf{y} = 0, \\ \mathbf{y} \geq 0, \end{aligned}$$

   with $A \in \mathbb{R}^{m\times d}$ for $m > d$, $\mathbf{b}, \mathbf{y} \in \mathbb{R}^m$. Show that if it is unbounded then we can find a feasible point $\mathbf{y}'$ such that

   (a) $\mathbf{b}^T\mathbf{y}' < 0$

   (b) $\mathbf{y}'$ has at least $m - (d + 1)$ zero components.

   Hint: use the simplex method.

2. Prove that if m halfspaces in $\mathbb{R}^d$, with $m > d$, do not have a point in common, then there exist some $d + 1$ of them that do not have a point in common, either.

   Hint: model m halfspaces in $\mathbb{R}^d$, with $m > d$ and with no points in common with a linear problem (P). Is it feasible/infeasible? Bounded/unbounded? What is the dual problem? Use this to select a linear subproblem of (P) to model $d + 1$ halfspaces without a point in common.

Exercise 6 - Load Balancing problem (***) Consider a collection of n jobs that must be executed and m machines, $M_1, \ldots, M_m$. Executing job j on any of the machines takes time t_j, where $t_j > 0$. The aim is to assign the jobs to the machines in such a way that the so-called makespan, i.e. the time until all jobs are finished, is as small as possible. Let's denote the collection of jobs assigned to machine M_i by $A(i)$. Then the load T_i of machine M_i, the total time for which M_i is busy, is given by

$$T_i = \sum_{j \in A(i)} t_j,$$

and the makespan of the assignment is $T := \max_{1 \leq i \leq m} T_i$. Assume that each job is assigned to a single machine. Load Balancing is NP-hard. Consider the following greedy approximation algorithm: we consider the jobs one by one and assign each job to the machine whose current load is smallest.

Greedy algorithm for Load Balancing $(t_1, \ldots, t_n, m)$

1. Initialize $T_i = 0$ and $A(i) = \emptyset$ for $1 \leq i \leq m$
2. For $j = 1, \ldots, n$
   (a) Find a k such that $T_k = \min_{1 \leq i \leq m} T_i$
   (b) Assign job j to the machine M_k of minimum load
   (c) $A(k) = A(k) \cup \{j\}$ and $T_k = T_k + t_j$

---

1. **Optional**: What data structure can you choose to implement the algorithm in order to have it run in polynomial time? Why?
2. Explain why the optimal value of the problem OPT is such that $OPT \geq LB := \max(\frac{1}{m} \sum_{1 \leq j \leq n} t_j, \max_{1 \leq j \leq n} t_j)$.
3. Prove that the algorithm is a 2-approximation algorithm, i.e. prove that it produces an assignment of jobs to machines such that the makespan T satisfies $T \leq 2OPT$. Hint: use the lower bound.

Optional, extra points: How to improve this algorithm? A weak point of this algorithm is the following. Suppose to first have a large number of small jobs and then finally a single very large job. The algorithm will first spread the small jobs evenly over all machines and then add the large job to one of these machines. It would have been better, however, to give the large job its own machine and spread the small jobs over the remaining machines. Note that our algorithm would have produced this assignment if the large job would have been handled first. This observation suggest the following adaptation of the greedy algorithm: we first sort the jobs according to decreasing processing times, and then run the previous procedure.

4. Consider a set of n jobs with processing times $t_1, \ldots, t_n$ that have to be scheduled on m machines, where $t_1 \geq t_2 > \cdots \geq t_n$. Prove that if $n > m$, then $OPT \geq t_m + t_{m+1}$.
5. Prove that the new algorithm is a (3/2)-approximation algorithm.