ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Department of Computer Science
Institute of Theoretical Computer Science
Bernd Gärtner and Niao He

Optimization for Data Science — Graded Assignment 2 — FS23

- The solution is due on **May 1, 2023** by **11:59 pm**. Please submit your solution as a PDF on Moodle. The name of the file should follow the format `GA4-{Legi number}`, e.g., `GA2-19-123-456`. After uploading your solution, please make sure that the status is "Submitted for grading". You should receive an automatic email that confirms your submission. Please notify us if you don't receive this.
- If you want to submit your solution within six hours before the deadline and a technical problem prevents you from submitting it on Moodle, you can send your solution as PDF to `saeed.ilchi@inf.ethz.ch`. The same submission deadline still applies. If you encounter any trouble with the submission process, complain timely.
- Please solve the exercises carefully and typeset your solution using LaTeX or a similar typesetting program. A tutorial can be found at `http://www.cadmo.ethz.ch/education/thesis/latex`. Handwritten solutions will not be graded! The same applies to solutions written with any kind of tablet device and stylus, etc.
- For geometric drawings that can easily be integrated into LaTeX documents, we recommend the drawing editor IPE, which you can find at `http://ipe7.sourceforge.net/`.
- Keep in mind the following premises:
  - When writing in English, write short and simple sentences.
  - When writing a proof, write precise statements.
- This is a theory course, which means: if an exercise does not explicitly say "you do not need to prove your answer" or "justify intuitively", then a formal proof is **always** required. You can of course refer in your solutions to the lecture notes and to the exercises, if a result you need has already been proved there.
- We would like to stress that the ETH Disciplinary Code applies to this Graded Assignment as it constitutes part of your final grade. The only exception we make to the Code is that we encourage you to verbally discuss the tasks with your colleagues. It is strictly prohibited to share any (hand-)written or electronic (partial) solutions with any of your colleagues. We are obliged to inform the Rector of any violations of the Code.
- As with all exercises, the material of the graded assignments is relevant for the exam.

CGD with Unknown Smooth Parameter (25 points)

Suppose a function $f : \mathbb{R}^d \to \mathbb{R}$ (for $d \geqslant 2$) that is differentiable and coordinate-wise smooth with parameter $\mathcal{L} = (L_1, L_2, \ldots, L_d)$ where exactly one of L_j-s is equal to β ($\beta > 1$) and the others are all equal to 1. Moreover, suppose that f is μ-strongly convex ($\mu \leqslant 1$). Now we know β and μ, but we do not know which coordinate is β-smooth. We consider Algorithm 1: starting with a guess $\tilde{\mathcal{L}}^{(0)} = (\tilde{L}_1 = 1, \tilde{L}_2 = 1, \ldots, \tilde{L}_d = 1)$, when a coordinate-wise sufficient decrease criterion

$$f\left(x_t - \frac{1}{\tilde{L}_i^{(t)}} \nabla_i f(x_t) e_i\right) \leqslant f(x_t) - \frac{1}{2\tilde{L}_i^{(t)}} \|\nabla_i f(x_t)\|^2 \tag{1}$$

is not satisfied, we update our guess of the smoothness parameter of this coordinate to be β. In Algorithm 1, $\mathcal{D}_{IS}(L_1, L_2, \ldots, L_d)$ represents the probability distribution with the following mass function:

$$\mathbb{P}[i = k] = \frac{L_k}{\sum_{j=1}^d L_j}, \quad \forall k \in [d].$$

Algorithm 1 FUNNY-CGD $(x_0, \varepsilon, \beta, \mu)$

$\tilde{\mathcal{L}}^{(0)} := \left(\tilde{L}_1^{(0)}, \tilde{L}_2^{(0)}, \ldots, \tilde{L}_d^{(0)}\right) = (1, 1, \ldots, 1)$
$T = \left\lceil \frac{2(\beta+(d-1))}{\mu} \ln \frac{1}{\varepsilon} \right\rceil$
for $t = 0, 1, 2, \ldots, T-1$ **do**
  Sample i from $\mathcal{D}_{IS}\left(\tilde{L}_1^{(t)}, \tilde{L}_2^{(t)}, \ldots, \tilde{L}_d^{(t)}\right)$
  if $f\left(x_t - \frac{1}{\tilde{L}_i^{(t)}} \nabla_i f(x_t) e_i\right) \leqslant f(x_t) - \frac{1}{2\tilde{L}_i^{(t)}} \|\nabla_i f(x_t)\|^2$ **then**
    $\tilde{\mathcal{L}}^{(t+1)} = \tilde{\mathcal{L}}^{(t)}$
    $x_{t+1} = x_t - \frac{1}{\tilde{L}_i^{(t)}} \nabla_i f(x_t)\, e_i$
  else
    $\tilde{\mathcal{L}}^{(t+1)} = \left(\tilde{L}_1^{(t)}, \ldots \tilde{L}_{i-1}^{(t)}, \beta, \tilde{L}_{i+1}^{(t)}, \ldots, \tilde{L}_d^{(t)}\right)$
    $x_{t+1} = \text{CGD-RAND}\left(x_t, \left\lceil \frac{\beta+(d-1)}{\mu} \ln 2 \right\rceil, \mathcal{D}_{IS}\left(\tilde{\mathcal{L}}^{(t+1)}\right), \gamma\right)$ with
γ equals to $\left(\frac{1}{\tilde{L}_1^{(t+1)}}, \ldots, \frac{1}{\tilde{L}_d^{(t+1)}}\right)$
  end if
end for
return x_T

Algorithm 2 CGD-RAND $(x_0, T, \mathcal{D}, \gamma = \{\gamma_1, \gamma_2, \ldots, \gamma_d\})$

for $t = 0, 1, 2, \ldots, T-1$ **do**
  Sample i from Distribution $\mathcal{D}$
  $x_{t+1} = x_t - \gamma_i \nabla_i f(x_t)\, e_i$
end for
return x_T

Prove the followings for Algorithm 1.

(a) Show that when Algorithm 1 stops, it queries at most $O\left(\frac{d\bar{L}}{\mu}\ln\frac{1}{\varepsilon}\right)$ numbers of partial derivatives of f with $\bar{L}=\frac{1}{d}\sum_{j=1}^{d}L_j$.

(b) Show that the output from Algorithm 1 satisfies

$$\mathbb{E}\left[f\left(x_T\right)-f\left(x^*\right)\right]\leqslant\varepsilon\left(f\left(x_0\right)-f\left(x^*\right)\right).$$

Normalized GD for Nonconvex Optimization (25 points)

Consider a L-smooth function $f:\mathbb{R}^d\to\mathbb{R}$, which could be nonconvex. In addition, we assume the function is differentiable and has a global minimum x^*. Our goal is to find a stationary point x such that $\|\nabla f(x)\|$ is small. Instead of using conventional gradient descent, we consider a normalized version as follows:

$$x_{t+1}=x_t-\eta_t\frac{\nabla f(x_t)}{\|\nabla f(x_t)\|+\beta_t},$$

where $\eta_t,\beta_t>0$.

(a) In this part, we consider fixed η_t and β_t, i.e., $\eta_t=\eta$ and $\beta_t=\beta$ for $t\geqslant 0$. Find a stepsize η such that $f(x_{t+1})\leqslant f(x_t)$ for all iterations.

(b) Under the same setting as part (a), show that the algorithm converges with rate $\mathcal{O}\left(T^{-1}\right)$, i.e.,

$$\frac{1}{T}\sum_{t=0}^{T-1}\|\nabla f(x_t)\|^2=\mathcal{O}\left(T^{-1}\right).$$

(c) Now we consider the more general case where η_t and β_t are allowed to change over time. Design η_t and β_t such that without knowing the smoothness parameter, i.e., η_t and β_t should not depend on L, the algorithm provides the following guarantee:

$$\frac{1}{T}\sum_{t=0}^{T-1}\|\nabla f(x_t)\|=\tilde{\mathcal{O}}\left(T^{-1/2}\right).$$

where $\tilde{\mathcal{O}}(\cdot)$ hides logarithmic terms of T. Note that now we bound the average of gradient norms (instead of squared norms), and that is why the right-hand side depends on $\sqrt{T}$ rather than T.

Frank-Wolfe with an Approximation Oracle (20 points)

Recall that in the Frank-Wolfe algorithm, we assume there is a linear minimization oracle (LMO) that can return the exact minimizer. However, this minimization problem can itself be challenging to solve. In this exercise, we analyze the convergence of a variant of the Frank-Wolfe algorithm with an approximation LMO.

We consider the optimization problem $\min_{x \in X} f(x)$ for a convex function $f : \mathbb{R}^d \to \mathbb{R}$ with smooth parameter 1, and for X equals to $[-1/2, 1/2]^d$ (i.e., X is a unit d-dimensional cube). We assume a minimizer $x^* \in X$ exists. For any accuracy $\alpha \geqslant 0$, let $\text{APPROX-LMO}_X^\alpha(g)$ be an oracle that computes a vector in X such that

$$\nabla f(x_t)^\top \text{APPROX-LMO}_X^\alpha(g) \leqslant \min_{z \in X} \nabla f(x_t)^\top z + \alpha C_{(f,X)}$$

where $C_{(f,X)}$ is the curvature constant. So for α being zero, APPROX-LMO_X^0 is an exact oracle.

Algorithm 3 shows the modified Frank-Wolfe algorithm,

Algorithm 3 $\text{APPROX-FW}\ (x_0, T)$

for $t = 0, 1, 2, \ldots, T-1$ **do**
  $\gamma_t = \frac{2}{t+2}$
  $s_t = \text{APPROX-LMO}_X^{\gamma_t}(\nabla f(x_t))$
  $x_{t+1} = (1-\gamma_t)x_t + \gamma_t s_t$
end for
return x_T

(a) Show that

$$f(x_{t+1}) \leqslant f(x_t) - \gamma_t g(x_t) + \gamma_t^2 d \tag{2}$$

where $g(x_t)$ is the duality gap that is defined in the lectures.

(b) Show that for any $\varepsilon > 0$, and for any $T \geqslant 4d/\varepsilon$, we have:

$$f(x_T) - f(x^*) \leqslant \varepsilon.$$

Modified Newton's Method (30 points)

Let $f : \mathbb{R}^d \to \mathbb{R}$ be a convex twice-differentiable function with Lipschitz Hessian. In order to minimize the given function we consider a modified version of Newton's Method.

Algorithm 4 MODIFIED-NEWTON (x_0, H, T)

for $t = 0, 1, 2, \ldots, T-1$ **do**
  $\lambda_t = \sqrt{H\|\nabla f(x_t)\|}$
  $x_{t+1} = x_t - (\nabla^2 f(x_t) + \lambda_t I)^{-1} \nabla f(x_t)$
end for
return x_T

Assume the following for this exercise:

1. There is a $x^* \in \mathbb{R}^d$ such that $f(x^*) = \min_{x \in \mathbb{R}^d} f(x)$.
2. There is a constant $B \in \mathbb{R}$ such that for all $x \in \mathbb{R}^d$, if $f(x) \leqslant f(x_0)$, then $\|x - x^*\| \leqslant B$.
3. There is a constant $H > 0$ such that for all $x, y \in \mathbb{R}^d$, we have:

$$f(y) \leqslant f(x) + \nabla f(x)^\top (y - x) + \frac{1}{2}(y - x)^\top \nabla^2 f(x)(y - x) + \frac{H}{3}\|y - x\|^3$$

and

$$\|\nabla f(y) - \nabla f(x) - \nabla^2 f(x)(y - x)\| \leqslant H\|y - x\|^2.$$

In the following steps, we derive a convergence result for Algorithm 4 given the three assumptions above on f.

(a) Show that the following relations holds for all λ_t in Algorithm 4:

$$\lambda_t(x_{t+1} - x_t) = -\nabla f(x_t) - \nabla^2 f(x_t)(x_{t+1} - x_t). \tag{3}$$

(b) Show that for all iterations, the followings hold:

$$H\|x_{t+1} - x_t\| \leqslant \lambda_t,$$
$$\|\nabla f(x_{t+1})\| \leqslant 2\lambda_t\|x_{t+1} - x_t\| \leqslant 2\|\nabla f(x_t)\|.$$

(c) Prove the following descent lemma for Algorithm 4:

$$f(x_{t+1}) \leqslant f(x_t) - \frac{2}{3}\lambda_t\|x_{t+1} - x_t\|^2.$$

(d) Let $\mathcal{I}_\infty = \{i \in \mathbb{N} : \|\nabla f(x_{i+1})\| \geqslant \frac{1}{4}\|\nabla f(x_i)\|\}$ be the set of iterations at which the norm of gradient shrinks by at least a factor four.

Show that for all $t \in \mathcal{I}_\infty$, we have:

$$f(x_{t+1}) - f(x_t) \leqslant -\frac{1}{96B^{3/2}\sqrt{H}}(f(x_t) - f(x^*))^{3/2}.$$

Hint: First show that for any iteration, we have $f(x_t) - f(x^*) \leqslant B\|\nabla f(x_t)\|$.

Remark: Using the properties above, we can show that for all iterations (note that this holds for all t and not just members of $\mathcal{I}_\infty$):

$$f(x_t) - f(x^*) = \mathcal{O}(1/t^2).$$

Let us emphasize that proving this bound is not part of the exercise.