S&DS431/631 Computation and Optimization **out:** Nov 9, 2023

Problem Set 5

Lecturer: Zhuoran Yang **due:** Nov 30, 2023

Homework Policy

You should submit your homework using ***Gradescope*** via Canvas. It is due by ***11:59pm Eastern Time on the due date***. We *will not* accept late submissions.

Submit your work in the form of **a single PDF file** unless otherwise required. If you are required to submit your code, we will make it clear whether you need to submit the code files separately or print them in your PDF submission (e.g., using latex).

- If you write on a Tablet and export your solutions in PDF, or write on paper and scan your solutions as a PDF file, it would be great if
- If you write in LaTex, try to write the equations in a good style; define the notations and symbols before you use them.

To print Python code in LaTeX, you can use `verbatim`.

You are allowed to discuss the problems in groups of up to three (you and up to two others) you must *write up the solutions on your own.* If you do work with anyone, you should acknowledge your collaborators. Similarly, you must cite all references that you use other than the lecture notes for the course (you may not search the web for answers, however).

[BV]: In the following, [BV] stands for the book *Convex Optimization* by Boyd and Vandenberghe. Link: `https://web.stanford.edu/~boyd/cvxbook/bv_cvxbook.pdf`

[AE]: In the following, [AE] stands for the *Additional Exercises for Convex Optimization* by Boyd and Vandenberghe. Link: `https://github.com/cvxgrp/cvxbook_additional_exercises/blob/main/additional_exercises.pdf`

Problem 1: Exercise of 9.7 [BV] Normalized and Unnormalized steepest descent directions. (30 Points)

Given a norm $\|\cdot\|$ defined on $\mathbb{R}^n$, its dual norm is defined as

$$\|v\|_* = \arg \sup_{u \in \mathbb{R}^n} \{\langle u, v\rangle : \|u\| \leq 1\}.$$

See Section 9.4 of [BV] for the definitions of the steepest descent direction and its normalized version.

Problem 2: Exercise 9.1 (b) of [AE] Gradient descent and non-differentiable functions. Only solve question (b). Here $\gamma > 1$. (30 Points)

Problem 3: Proximal Gradient Descent (I) (40 Points)

We introduced the proximal gradient algorithm for minimizing a sum of functions:

$$\min_{x\in\mathbb{R}^n} f(x) = g(x) + h(x),$$

where $g : \mathbb{R}^n \to \mathbb{R}$ is a continuously differentiable function and $h : \mathbb{R}^n \to \mathbb{R}$ is a "simple" convex function. The proximal operator induced by h is defined as

$$\text{prox}_{\alpha,h}(x) = \arg\min_z \left\{ \frac{1}{2\alpha}\|x - z\|_2^2 + h(z) \right\}. \tag{1}$$

The proximal gradient updates (with a constant stepsize α) are given by

$$z^{k+1} = x^k - \alpha \cdot \nabla g(x^k)$$
$$x^{k+1} = \text{prox}_{\alpha,h}(z^{k+1}).$$

We assume that g is *smooth* in the sense that ∇g is M-Lipschitz continuous, i.e.,

$$\|\nabla g(x) - \nabla g(y)\|_2 \le M \cdot \|x - y\|_2, \qquad \forall x, y \in \mathbb{R}^n.$$

Questions:

(a) For any $x, y \in \mathbb{R}^n$, define

$$U(x, y) = g(x) + \langle \nabla g(x), y - x\rangle + \frac{1}{2\alpha} \cdot \|y - x\|_2^2 + h(y). \tag{2}$$

Note that $U(x, y)$ is $1/\alpha$-strongly convex in y. Prove that

$$\arg\min_{y\in\mathbb{R}^n} U(x, y) = \text{prox}_{\alpha,h}\big(x - \alpha \cdot \nabla g(x)\big) \tag{3}$$

for any $x \in \mathbb{R}^n$.

(b) In particular, let $\mathcal{C}$ be a convex set and consider one iteration of projected gradient descent for solving $\min_{x\in\mathcal{C}} g(x)$:

$$y = \Pi_{\mathcal{C}}\big(x - \alpha \cdot \nabla g(x)\big),$$

where $\Pi_{\mathcal{C}}$ is the projection operator to set $\mathcal{C}$ and α is the stepsize. Use (a) to prove that y is a minimizer of the following problem:

$$\min_v \; F(v) = f(x) + \langle \nabla f(x), v - x\rangle + \frac{1}{2\alpha}\|v - x\|_2^2, \qquad \text{subject to} \;\; v \in \mathcal{C}.$$

(c) Suppose $\alpha \le 1/M$, for any $x, y \in \mathbb{R}^n$, prove that $U(x, y) \ge f(y) = g(y) + h(y)$.

(d) Now suppose g is smooth and possibly nonconvex. We prove that with $\alpha = 1/M$, proximal gradient converges to a stationary point on average. Here we have a possibly nonsmooth regularizer h and thus the notion of *stationary point* needs to be modified. In particular, let's define a mapping $G_\alpha : \mathbb{R}^n \to \mathbb{R}^n$:

$$G_\alpha(x) = \frac{1}{\alpha}\Big(x - \mathrm{prox}_{\alpha,h}\big(x - \alpha \cdot \nabla g(x)\big)\Big). \tag{4}$$

If x^* is a stationary point of proximal gradient descent, i.e., if $x^* = \mathrm{prox}_{\alpha,h}\big(x^* - \alpha \cdot \nabla g(x^*)\big)$, we have $G_\alpha(x^*) = 0$. Thus, G_α can be viewed as a surrogate of ∇f when f is not differentiable (and h is convex). Now suppose both h and g are differentiable, prove that for any x^* satisfying $G_\alpha(x^*) = 0$, we have $\nabla f(x^*) = 0$, i.e., x^* is a stationary point of f. *Hint*: When h is differentiable, $\mathrm{prox}_{\alpha,h}(x)$ satisfies a first-order optimality condition.

Problem 4: Proximal Gradient Descent (II) (40 Points)

Now we study the convergence of proximal gradient descent for objective function $f = g + h$, where g is *nonconvex* and *M-smooth*, and h is convex. To simplify the analysis, we **further assume h is differentiable**. Handling nondifferentiable h requires a tool called *subdifferential*, which will be introduced in S&DS 432.

Recall that we define the mapping G_α in (4). Using this notation, we can write the proximal gradient updates as

$$x^{k+1} = x^k - \alpha \cdot G_\alpha(x^k). \tag{5}$$

This form resembles standard gradient descent. Besides, we can equivalently write

$$x^{k+1} = \arg\min_y U(x^k, y).$$

Questions:

(a) Using the optimality condition of x^{k+1}, prove that

$$\frac{1}{\alpha}\Big[x^{k+1} - \big(x^k - \alpha \cdot \nabla g(x^k)\big)\Big] + \nabla h(x^{k+1}) = 0. \tag{6}$$

Equivalently, we have $\nabla g(x^k) - G_\alpha(x^k) + \nabla h(x^{k+1}) = 0$.

(b) Use the convexity of h to prove that

$$h(x^{k+1}) + \alpha \cdot \langle \nabla h(x^{k+1}), G_\alpha(x^k) \rangle \le h(x^k).$$

(c) Use (a) and (b) above and Question (c) of Problem 3 to conclude the following descent lemma:

$$f(x^{k+1}) \leq U(x^k, x^{k+1}) \leq f(x^k) - \frac{\alpha}{2}\|G_\alpha(x^k)\|_2^2.$$

(No need to prove the following argument.) Thus, similar to the proof of gradient descent, by setting $\alpha = 1/M$, we can prove that

$$\sum_{k=1}^{K} \|G_\alpha(x^k)\|_2^2 \leq \frac{2M}{K} \cdot \big(f(x^1) + f^*\big).$$

Problem 5: Monotonicity of Proximal Mapping. (20 Points)

Let h be a differentiable and convex function. The proximal operator $\text{prox}_{\alpha,h}$ is defined in (1). Let $\text{prox}_h(\cdot)$ denote prox_h with $\alpha = 1$ for simplicity. Let $x_1, x_2 \in \mathbb{R}^n$ be any two vectors. Let $z_1 = \text{prox}_h(x_1)$ and $z_2 = \text{prox}_h(x_2)$.

Questions:

(a) Prove that

$$\langle \text{prox}_h(x_1) - \text{prox}_h(x_2), x_1 - x_2 \rangle \geq \|\text{prox}_h(x_1) - \text{prox}_h(x_2)\|_2^2.$$

Hint: Use Exercise 3.11 in [BV].

(b) Use (a) to prove that

$$\|\text{prox}_h(x_1) - \text{prox}_h(x_2)\|_2 \leq \|x_1 - x_2\|_2.$$

Problem 6: Sparse linear regression with proximal gradient and projected gradient. (40 Points)

Assume we have a dataset $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_m, y_m)$ where $\mathbf{x}_i \in \mathbb{R}^n$ are sampled i.i.d. from a Gaussian distribution $N(\mathbf{0}, \mathbf{I}_n)$ and $y_i \in \mathbb{R}$ are the corresponding binary labels. We assume a linear regression model

$$y_i = \mathbf{x}_i^\top \beta^* + \epsilon_i,$$

where $\epsilon_i \sim N(0, \sigma^2)$ is a Gaussian noise. Using matrix notation, we can write

$$\mathbf{y} = \mathbf{X}\beta^* + \epsilon.$$

Moreover, when the dimension d is large, we often assume that the true parameter β^* is sparse, i.e., most of the entries of β^* are zero. Our goal is to estimate β^* from the data. (The setting is the same as that of the last problem in the previous problem set.)

The least-squares loss function is:

$$\ell(\beta) = \|\mathbf{y} - \mathbf{X}\beta\|_2^2 = \sum_{i=1}^{m} |y_i - \mathbf{x}_i^\top \beta|^2.$$

Our goal is to find the optimal sparse parameter vector $\beta^* \in \mathbb{R}^n$ by minimizing the above loss function using a proximal gradient method.

We consider three methods: **(i) convex optimization via CVX** and **(ii) proximal gradient descent**, and **(iii) projected gradient descent**, **(iv) Frank-Wolfe method**.

(i) Convex optimization via CVX

The objective function is

$$L_\lambda(\beta) = \ell(\beta) + \lambda \cdot \|\beta\|_1$$

and we minimize this function without a constraint. The ℓ_1-norm regularization induces sparsity in the solution. This method is already implemented. We also implement the constrained version

$$\min \ell(\beta) \qquad \text{subject to} \quad \|\beta\|_1 \leq \lambda.$$

which is compared with Frank-Wolfe.

(ii) Proximal gradient descent

We also consider using proximal gradient to minimize $L_\lambda(\beta)$. For $h(\beta) = \lambda \cdot \|\beta\|_1$, the proximal mapping $\text{prox}_{\alpha,h}$ is given by the soft-thresholding function

$$S_\tau(t) = \begin{cases} 0 & \text{if } |t| \leq \tau \\ t - \tau & \text{if } t > \tau \\ -t + \tau & \text{if } t < -\tau. \end{cases} \tag{7}$$

One step of proximal gradient is given by

$$\beta \leftarrow S_{\alpha\cdot\lambda}\big(\beta - \alpha \cdot \nabla \ell(\beta)\big),$$

where the soft-thresholding operator with $\tau = \alpha\cdot\lambda$ is applied to each entry of the vector $\beta - \alpha\cdot\nabla\ell(\beta)$.

(iii) Projected gradient descent

Now we consider another way to estimate β^*. The optimization problem is given by

$$\min_\beta \ell(\beta) \qquad \text{subject to} \quad \|\beta\|_0 \le s, \tag{8}$$

where s is an integer. Here $\|v\|_0$, the ℓ_0-norm, counts the number of nonzero entries of a vector v, i.e., $\|v\|_0 = \sum_{j=1}^n \mathrm{I}\{v_j \neq 0\}$. This is a nonconvex optimization problem because the constraint set

$$\mathcal{C} = \{v \colon \|v\|_0 \le s\}$$

is nonconvex. But we can still define the projection operator $\Pi_{\mathcal{C}}$. The projection operator admits a closed-form expression as follows.

Question:

(a) For any vector $v \in \mathbb{R}^n$, we let $\mathcal{V} = \{1, \ldots, n\}$ to denote the indices of the largest s entries in absolute value. In other words, $\mathcal{V}$ has cardinality s, and for any $j \in \mathcal{V}$ and $j' \notin \mathcal{V}$, $|v_j| \ge |v_{j'}|$. Prove that $v' = \Pi_{\mathcal{C}}(v)$ satisfies $v'_j = v_j$ if $j \in \mathcal{V}$, and $v'_j = 0$ if $j \notin \mathcal{V}$.

With this projection operator, we can implement the projected gradient algorithm.

(iv) Frank-Wolfe Method Frank-Wolfe algorithm targets at

$$\min \ell(\beta) \qquad \text{subject to} \quad \|\beta\|_1 \le \lambda.$$

We only implement a constant-stepsize version. In each iteration, we need to solve the update direction $d^t = \mathrm{update}(g^t, \lambda)$, which is defined as the solution to

$$\min_d \langle d, g^t \rangle \qquad \text{subject to} \quad \|d\|_1 \le \lambda.$$

Here $g^t = \nabla\ell(\beta^t)$ is the current gradient. Then we have $\beta^{t+1} = (1-\alpha)\cdot\beta^t + \alpha \cdot d^t$.

Question:

(b) Complete the python functions that implement (a) the Soft-thresholding operator, (b) the projection operator, (c) the update direction of Frank-Wolfe. Include the code in your solutions.

(c) Implement these algorithms and export the output of the notebook. You should expect to see that all these three methods accurately recover β^*. Perhaps the surprising part is that the projected gradient descent also works, even when the set is nonconvex. This generally is not true.