Homework 2

CMU 10-725: Convex Optimization

OUT: Tuesday, Feb 3rd, 2025

DUE: Thursday, Feb 24th, 2025, 11:59pm

START HERE: Instructions

- **Collaboration policy:** Collaboration on solving the homework is allowed, after you have thought about the problems on your own. To remind you, many questions in this HW have solutions that are very easy to find online (and many are from previous versions of this course). It is also OK to get clarification (but not solutions) from books or online resources, again after you have thought about the problems on your own. There are two requirements: first, cite your collaborators fully and completely (e.g., "Jane explained to me what is asked in Question 2.1"). Second, write your solution *independently*: close the book and all of your notes, and send collaborators out of the room, so that the solution comes from you only.
- **Submitting your work:**
  - **Gradescope:** For the written problems such as short answer, multiple choice, derivations, proofs, or plots, we will be using the Gradescope. The best way to format your homework is by using the Latex template released in the handout and writing your solutions in Latex. However, submissions can be handwritten onto the template, but should be labeled and clearly legible. If your writing is not legible, you will not be awarded marks.
    Regrade requests can be made after the homework grades are released, however this gives the TA the opportunity to regrade your entire paper, meaning if additional mistakes are found then points will be deducted.
  - **Programming:** You should submit all code used to solve the programming aspect of the homework to the corresponding 'Programming' submission slot on Gradescope. If you do not do this, you will not get any credit for any of the programming section irrespective of the plots and values submitted to the 'Written' submission slot.

1 Implications of Smoothness and Strong Convexity (20 points) - Julia

Let f be convex and twice differentiable.

1. **(10 pts)** Show that the following statements are equivalent.

   (a) ∇f is Lipschitz with constant β;

   (b) $(\nabla f(x) - \nabla f(y))^T(x-y) \leq \beta\|x-y\|_2^2$ for all x, y;

   (c) $\nabla^2 f(x) \preceq \beta I$ for all x;

   (d) $f(y) \leq f(x) + \nabla f(x)^T(y-x) + \frac{\beta}{2}\|y-x\|_2^2$ for all x, y.

   We suggest that you prove (a) $\Rightarrow$ (b), (b) $\Rightarrow$ (c), (c) $\Rightarrow$ (d), (d) $\Rightarrow$ (b), and (c) $\Rightarrow$ (a). You are welcome to try alternative approaches, but your response will be graded with respect to these 5 parts.

   Hints: You can use the following property of the directional derivative of $g : \mathbb{R}^n \to \mathbb{R}$ at h, where $\|h\|_2 = 1$:

   $$\nabla g(x)^T h = \lim_{t \to 0} \frac{g(x+th) - g(x)}{t}.$$

   As an application of this, we have that

   $$\nabla^2 f(x) h = \lim_{t \to 0} \frac{\nabla f(x+th) - \nabla f(x)}{t}.$$

   Also recall the Taylor expansion with Lagrange form for the remainder:

   $$f(y) = f(x) + \nabla f(x)^T(y-x) + \frac{1}{2}(y-x)^T \nabla^2 f(\lambda x + (1-\lambda)y)(y-x).$$

   Finally, recall the integral form of the mean-value theorem: we have that for all x, y:

   $$\nabla f(y) - \nabla f(x) = \int_0^1 \nabla^2 f(x + t(y-x))(y-x)\, dt.$$

   Solution To show $(a) \implies (b)$, we have

   $$||\nabla f(x) - \nabla f(y)|| \leq \beta ||x - y||.$$

   Recall

.

2. **(10 pts)** Show that the following statements are equivalent.

(a) f is strongly convex with constant α;

(b) $(\nabla f(x) - \nabla f(y))^T(x-y) \geq \alpha\|x-y\|_2^2$ for all x, y;

(c) $\nabla^2 f(x) \succeq \alpha I$ for all x;

(d) $f(y) \geq f(x) + \nabla f(x)^T(y-x) + \frac{\alpha}{2}\|y-x\|_2^2$ for all x, y.

We suggest that you prove (a) $\Rightarrow$ (b), (b) $\Rightarrow$ (c), (c) $\Rightarrow$ (d), and (d) $\Rightarrow$ (a). You are welcome to try alternative approaches, but your response will be graded with respect to these 4 parts.

Solution YOUR SOLUTION HERE

2 Projection to the PSD Cone (6 Pts) - Johnna

In this problem, we study how to project a given point onto a convex set. These kinds of projections will be used in projected gradient descent algorithms.

The $d \times d$-dimensional PSD cone $\mathcal{P}$ is given by:

$$\mathcal{P} = \{M \in \mathbb{R}^{d \times d} \mid M \succeq 0\}.$$

Here, $M \succeq 0$ means that M is a symmetric, positive semi-definite matrix. Derive (with proof) a formula for $\Pi_{\mathcal{P}}(X) \doteq \text{argmin}_{M \in \mathcal{P}} \|M - X\|_F^2$, where $X \in \mathbb{R}^{d \times d}$ is a symmetric matrix (not necessarily PSD) and $\|\cdot\|_F^2$ denotes the Frobenius norm. You can use $X = U\Sigma U^\top$ as the eigen-decomposition of X, where U consists of the eigenvectors and Σ consists of the associated eigenvalues.

Solution YOUR SOLUTION HERE

3 Convex-Concave Functions and Saddle-Points (Canary, Zixin)

We say the function $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ is *convex-concave* if $f(x, z)$ is a concave function of z, for each fixed x, and a convex function of x, for each fixed z. [1] We also require its domain to have the product form $\text{dom}\, f = A \times B$, where $A \subseteq \mathbb{R}^n$ and $B \subseteq \mathbb{R}^m$ are convex.

(a) Give a second-order condition for a twice differentiable function $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ to be convex-concave, in terms of its Hessian $\nabla^2 f(x, z)$.

(b) Suppose that $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ is convex-concave and differentiable, with $\nabla f(\tilde{x}, \tilde{z}) = 0$. Show that the *saddle-point property* holds: for all x, z, we have

$$f(\tilde{x}, z) \leq f(\tilde{x}, \tilde{z}) \leq f(x, \tilde{z}).$$

[1] These types of functions arise in two-player zero sum games, where one player chooses x and the other chooses z, but they have opposite utilities, meaning that the X-player wants to minimize $f(x, z)$, while the Z player wants to maximize it. These functions also arise centrally in duality, encountered in the second half of the course.

Show that this implies that f satisfies the *strong max-min property*:

$$\sup_z \inf_x f(x,z) \;=\; \inf_x \sup_z f(x,z)$$

(and their common value is $f(\tilde{x},\tilde{z})$).

(c) Now suppose that $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ is differentiable, but not necessarily convex-concave, and the saddle-point property holds at $\tilde{x}, \tilde{z}$:

$$f(\tilde{x},z) \le f(\tilde{x},\tilde{z}) \le f(x,\tilde{z})$$

for all x, z. Show that $\nabla f(\tilde{x},\tilde{z}) = 0$.

Solution YOUR SOLUTION HERE

4 Step-Sizes and Optimization Algorithms (15 Pts) - Nuoya

4.1 Polyak Step-Size (10 Pts)

In our analysis of the subgradient method we began with the inequality:

$$\|x^{t+1} - x^*\|_2^2 \le \|x^t - x^*\|_2^2 - 2\eta_t(f(x^t) - f(x^*)) + \eta_t^2 \|g_{x^t}\|_2^2.$$

Boris Polyak, who was a giant in the field of mathematical optimization, proposed an "oracle" step-size rule which simply chooses η_t at each iteration to minimize the RHS above. This is known as the Polyak step-size, and requires one to know the value of $f(x^*)$ (i.e. the optimal function value) or at least a lower bound on this quantity.

This sequence of step-sizes has many magical properties (for instance, it automatically adapts to things like smoothness and strong-convexity, giving algorithms which are optimal in each of these settings, without needing to know the smoothness and strong-convexity parameters).

1. Calculate the Polyak step-sizes η_t above. Show that the Polyak step-sizes guarantee the following property:

$$\|x^{t+1} - x^*\|_2^2 \le \|x^t - x^*\|_2^2 - \frac{(f(x^t) - f(x^*))^2}{\|g_{x^t}\|_2^2}.$$

This in turn means that the Polyak step-sizes ensure that at each step we move closer to an optimal solution, i.e. every step makes some progress towards the optimal solution.

Solution

2. Let us suppose that we are optimizing a function which is G-Lipschitz. Give a (tight) upper bound (it should match the best rate we've given in lecture) for the subgradient method for $f(x^{\text{best}}) - f(x^*)$.

Solution

3. Suppose we do not know $f(x^*)$ exactly but we have a known lower bound ℓ with $\ell \leq f(x^*)$. Define the slack $\delta := f(x^*) - \ell \geq 0$. Run the Polyak rule using ℓ in place of $f(x^*)$ and show that after k iterations the best iterate

$$x^{\text{best}} := \arg \min_{0 \leq t \leq k-1} f(x^t)$$

satisfies the bound

$$f(x^{\text{best}}) - f(x^*) \;\leq\; \sqrt{\delta^2 \;+\; \frac{G^2 \|x^0 - x^*\|_2^2}{k}}\,.$$

Solution

4.2 Step-Size Schedule (5 Pts)

In one of the lectures, we analyzed the subgradient method. We proved that if function f is Lipschitz continuous, convex, and

$$\sum_{t=0}^{\infty} \eta_t = \infty,$$
$$\sum_{t=0}^{\infty} \eta_t^2 < \infty,$$

then the subgradient method converges to the optimum. We can also note that the step-size schedule of $\eta_t = 1/\sqrt{t}$ also leads to convergence, but it does not satisfy the above two conditions. Show that the following two conditions suffice to ensure convergence for the subgradient method:

$$\sum_{t=0}^{\infty} \eta_t = \infty,$$
$$\lim_{t \to \infty} \eta_t = 0.$$

Hint: Try to argue that under these conditions,

$$\lim_{T \to \infty} \frac{\sum_{t=0}^{T} \eta_t^2}{\sum_{t=0}^{T} \eta_t} = 0.$$

Solution

5 Effects of Learning Rate on Optimization Performance (25 Pts) - Michael

In this problem, we will further explore (stochastic) gradient descent, some failure modes, and the crucial role that learning rate plays.

5.1 Quadratic Bowl Function (15 Pts)

In this subpart, we will consider the following function (called the "quadratic bowl")

$$f(x, y) = ax^2 + by^2.$$

Two or three sentences for each of the following questions is sufficient.

For the questions below suppose that $a = 1, b = 2$.

1. What are the relevant convexity and smoothness properties of f? (eg: is it strictly/strongly convex, is it Lipschitz, etc.?) As a result, using the bound of gradient descent that we proved in class for this function family, what does the theory tell us the learning rate should be? Given this learning rate, what do we expect the convergence rate to be?

   Solution YOUR SOLUTION HERE

2. For a given (x, y), manually derive the gradient of f. Then, implement the `gd_f` function in `bowl.py`: gradient descent of f from any starting point $\boldsymbol{x_0}$ given any learning rate α

   Solution YOUR SOLUTION HERE

3. For $\boldsymbol{x_0} = (3, -2)$, find a fixed learning rate such that `gd_f` converges extremely quickly (~ 3 steps). For full credit, you need to be within 5% of the optimal value. Does the learning rate you found align with what the theory predicts should work well?

   Solution YOUR SOLUTION HERE

4. Plot (1) the level sets of f; (2) the trajectories of $\boldsymbol{x_t}$ overlaid on the level set plot. What do you observe? (some questions to think about: *what if we increased b to 20 and kept the same learning rate? how does the shape of the curve affect convergence of gradient descent assuming we use the "optimal" learning rate?*)

   Solution YOUR SOLUTION HERE

For the questions below suppose that $a = 1, b = 20$. And $\boldsymbol{x_0} = (3, -2)$

5. Plot the level sets of f now. How did the shape of the curve change?

   Solution YOUR SOLUTION HERE

6. Again, based on the theory we derived in class what should we set the learning rate to now? Is it a different learning rate from Q5.1.1? Why?

Solution YOUR SOLUTION HERE.

7. Find a learning rate where gradient descent converges (should take roughly 80 steps). Plot the trajectories of $\boldsymbol{x_t}$ over the level set again, what do you notice? Why could we not converge as quickly as before when $b = 2$. Does this agree with your intuitions in Q5.1.4?

   Solution YOUR SOLUTION HERE

Now for the finale, based on our analysis above, make a *very simple* modification to the gradient descent algorithm (write your implementation in `gd_f_better`) such that running your new optimization algorithm will allow us to converge to the optimum in less than 10 steps.

Describe the changes that you made and plot the trajectory of $\boldsymbol{x_t}$ on the level set plot of f.

Hints:

1. *Modify how gradient descent uses the learning rate.*
2. *The way you set your learning rate should not depend on the initial point,* $\boldsymbol{x_0}$.

Solution YOUR SOLUTION HERE

5.2 Lasso Regression and Overparameterized Problems (10 Pts)

In this question, we will further explore Lasso regression as discussed in class. Lasso not only has nice geometric properties (i.e. it is convex; differentiable almost everywhere), but also nice statistical ones[2].

In this subpart, we will explore using Lasso+subgradient descent to solve a real-world regression problem: predicting the value of a house, given some features (lot area, year built, number of floors, etc.).

The regression you will be trying to learn is "overparameterized" meaning that the system we are trying to learn is underdetermined (i.e. more features than data points; this type of setup is very common nowadays[3]). In this case, Lasso will be particularly helpful.

As a reminder, here is the Lasso loss function: given a dataset with features matrix $X \in \mathbb{R}^{n \times d}$ and labels vector $y \in \mathbb{R}^n$,

$$L(w) = \frac{1}{2n}\|Xw - y\|_2^2 + \alpha||w||_1$$

where $\alpha \geq 0$.

1. Derive the subgradient of L. Implement both in `house.py`.

   Solution YOUR SOLUTION HERE

2. Implement subgradient descent using $L, \partial L$.

[2] You can read more about this here: https://www.stat.cmu.edu/~larry/=stat705/Lecture22.pdf.
[3] https://openai.com/index/deep-double-descent/

3. Perform a simple grid search over both the learning rate and α. You should be able to hit $R^2 > 0.8$ on the test set.

   (a) Search over $\alpha \in \{10^{k-5}\}_{k=0}^{7}$ and

   $$\texttt{lr} \in \{\texttt{1e-5, 5e-5, 1e-4, 5e-4, 1e-3, 5e-3, 1e-2}\}$$

   (b) Plot test R^2 as a function of both α and `lr` (using a heatmap may be helpful here). Note that some combinations of these parameters will not converge.

   (c) Select the optimal hyperparameters from your grid search (optimal in terms of test R^2, you should be able to hit $R^2 > 0.75$ easily) and for this set of hyperparameters plot (1) both the training and test loss as a function of time; (2) the training and test R^2 as a function of time.

4. Look at your learned coefficients, which features are particularly predictive of the final sale price?

5. Another way we could have implemented a sparse regression is through L_0 regularization:

   $$\tilde{L}(w) = \frac{1}{2n}\|Xw - y\|_2^2 + \alpha \sum_{i=1}^{d} \mathbb{1}(|w_i| > 0).$$

   Show that for $\alpha > 0$, $\tilde{L}$ is not convex in w. [*Hint: the best way to do this is through a counterexample example with an explicit setting of X, y, it is* ***not*** *true that the sum of a convex function with a non-convex one is not convex.*]

   Solution YOUR SOLUTION HERE

6 Collaboration Questions

1. (a) Did you receive any help whatsoever from anyone in solving this assignment? **Solution** Yes / No.

   (b) If you answered 'yes', give full details (e.g. "Jane Doe explained to me what is asked in Question 3.4")

Solution

2. (a) Did you give any help whatsoever to anyone in solving this assignment? **Solution** Yes / No.

   (b) If you answered 'yes', give full details (e.g. "I pointed Joe Smith to section 2.3 since he didn't know how to proceed with Question 2")

Solution

3. (a) Did you find or come across code that implements any part of this assignment? **Solution** Yes / No.

   (b) If you answered 'yes', give full details (book & page, URL & location within the page, etc.).

Solution