Advanced Optimization (2024 Fall)
Homework #1

Student ID, Name, Email

November 1, 2024

Evaluation: There is a problem section (in total 5 problems, 270pts) and a bonus section (5pts), and your score is the sum of the problem section and the bonus section. The scoring method for the problem section is as follows: Problem 1 (70pts) is asked to solve. Choose 3 of the remaining 4 problems (each with 50pts) to finish. There are two options for the final score evaluation of the problem section:

1. (**recommended**) If you choose 4 problems (Problem 1 + 3 selected ones, totally 220pts), you can obtain the full score (200pts) once you achieve at least 200pts.
2. If you choose 4 problems (totally 220pts) *and finish the remaining one (50pts)*:
   (a) If you haven't achieved 200pts on the chosen 4 problems, back to Case 1.
   (b) If you obtain $(245 + X)$pts ($X \geq 0$), the final score will be $(200 + X)$pts.

Attention: You are requested to indicate selected problem ids clearly.

My selected problem ids: 1,x,x,x.

```
% replace x,x,x by selected ids (e.g., 2,3,4,5)
% x,x,x = 2,3,4 by default if not explicitly specified
```

1 [70pts] APG Analysis and Implementation

Consider the following unconstrained composite optimization:

$$\min_{\mathbf{x}\in\mathbb{R}^d} F(\mathbf{x}) \triangleq f(\mathbf{x}) + h(\mathbf{x}),$$

where both $f(\cdot)$ and $h(\cdot)$ are convex. The function $f(\cdot)$ is L-smooth, whereas $h(\cdot)$ is not.

Proximal Gradient (PG) updates as $\mathbf{x}_{t+1} = \mathcal{P}_L^h(\mathbf{x}_t) \triangleq \mathbf{prox}_{\frac{1}{L}h}(\mathbf{x}_t - \frac{1}{L}\nabla f(\mathbf{x}_t))$, where **prox** is the proximal mapping. PG achieves an $\mathcal{O}(1/T)$ convergence rate; however, this rate is suboptimal, analogous to the suboptimality of Gradient Descent (GD) in smooth optimization settings. A natural approach to improve the convergence rate in composite optimization is to extend Nesterov's Accelerated Gradient Descent (AGD) method, leading to the development of the Accelerated Proximal Gradient (APG) algorithm:

$$\mathbf{x}_{t+1} = \mathcal{P}_L^h(\mathbf{y}_t), \quad \mathbf{y}_{t+1} = \mathbf{x}_{t+1} + \beta_t(\mathbf{x}_{t+1} - \mathbf{x}_t),$$

where $\beta_t > 0$ is a time-varying weight of the "momentum" term $(\mathbf{x}_{t+1} - \mathbf{x}_t)$.

(1) **[15pts]** Try to design β_t and prove the convergence rate of APG:

$$F(\mathbf{x}_T) - F(\mathbf{x}^\star) \leq \mathcal{O}\left(\frac{1}{T^2}\right).$$

(2) **[10pts]** For Lipschitz convex functions, we know that the Gradient Descent (GD) algorithm achieves a convergence rate of $\mathcal{O}(1/\sqrt{T})$. Given the strong performance of the APG method, one may wonder if we can "hack" the APG to obtain a faster convergence rate for Lipschitz convex functions.

Specifically, for the convex and Lipschitz optimization $\min_{\mathbf{x}\in\mathbb{R}^d} h(\mathbf{x})$, we can rewrite it as a composite optimization $\min_{\mathbf{x}\in\mathbb{R}^d} f(\mathbf{x}) + h(\mathbf{x})$, where $f(\mathbf{x}) = 0$ is convex and 0-smooth function, satisfying the requirements of APG. As such, it seems that the result of APG directly implies an $\mathcal{O}(1/T^2)$ convergence rate for Lipschitz convex function $h(\mathbf{x})$. Even $L = 0$ may cause trouble in the proximal mapping, we can add a small ε to rectify the issue. Is this idea correct? Give your answer, and briefly provide the reason.

(3) **[5pts]** To further understand the APG, let us consider a practical application: background modeling from videos. Suppose we are given a data matrix $D \in \mathbb{R}^{m\times d}$, which is expected to be decomposed as

$$D = L_0 + S_0,$$

where $L_0 \in \mathbb{R}^{m\times d}$ has low rank and $S_0 \in \mathbb{R}^{m\times d}$ is sparse. For example, if the data matrix D represents a sequence of frames from a monitoring video, the background

variations L_0 can be modeled as a low-rank structure because of the correlation across frames, while moving foreground objects S_0 can be represented as sparse components.

To achieve this goal, we formulate the following optimization problem:

$$\min_{L,S\in\mathbb{R}^{m\times d}} \frac{1}{2}\|D-L-S\|_{\mathrm{F}}^2+\mu\|L\|_*+\lambda\|S\|_1, \tag{1.1}$$

where $\mu,\lambda>0$ are hyperparameters, $\|A\|_*=\sum_i\sigma_i(A)=\mathrm{tr}(\sqrt{A^\top A})$ denotes the nuclear norm to impose the low-rank requirement on the matrix A, and $\|A\|_1=\sum_{ij}|A_{ij}|$ denotes the ℓ_1-norm to impose the sparsity requirement on the matrix A.

To solve (1.1), we can convert it into a composite optimization problem, where the optimization variable is $\mathbf{X}\triangleq(X^L,X^S)\in\mathbb{R}^{m\times d}\times\mathbb{R}^{m\times d}$ and the corresponding composite functions are $f(\mathbf{X})=\frac{1}{2}\|D-X^L-X^S\|_{\mathrm{F}}^2$ and $h(\mathbf{X})=\mu\|X^L\|_*+\lambda\|X^S\|_1$. Now the optimization problem becomes

$$\min_{\mathbf{X}\in\mathbb{R}^{m\times d}\times\mathbb{R}^{m\times d}} f(\mathbf{X})+h(\mathbf{X}).$$

Note that both $f(\cdot)$ and $h(\cdot)$ are convex, and $f(\cdot)$ is L_f-smooth w.r.t. the $\|\cdot\|$ norm (i.e. $\|\mathbf{X}\|\triangleq\sqrt{\|X^L\|_{\mathrm{F}}^2+\|X^S\|_{\mathrm{F}}^2}$).

Compute the smoothness parameter L_f of $f(\cdot)$ in this problem.

(4) **[40pts]** We can further use the APG algorithm to solve the background modeling from the video problem. Implement the PG and APG algorithms, compare the loss curves of PG and APG and attach the figure here. Detailed instructions are available in the `AOpt-Lab1/AOpt-Lab1.ipynb` jupyter notebook. Please make sure to export the completed ipynb file as an HTML file. Ensure that your outputs can be seen in the HTML file, and *submit the HTML file along with your homework*.

Solution. Give your answers here. (中英文均可)

2 [50pts] Non-convex Opt for Smooth Functions

We consider the unconstrained non-convex optimization problem $\min_{\mathbf{x}\in\mathbb{R}^d} f(\mathbf{x})$, where we assume $f(\cdot)$ is L-smooth. In class, we consider the convergence rate to a minimum to evaluate algorithms' performances. However, for non-convex functions, finding an exact optimal point is often challenging. Thus, we instead focus on the convergence rate to an ε-stationary point. Formally, we call $\mathbf{x}$ an ε-stationary point if the following is satisfied:

$$\|\nabla f(\mathbf{x})\|_2 \leq \varepsilon.$$

In the subsequent subproblems, we will analyze the gradient descent (GD) algorithm, prove the $\mathcal{O}(1/\sqrt{T})$ convergence rate to an ε-stationary point with deterministic feedback, and the $\mathcal{O}(1/T^{1/4})$ convergence rate with stochastic feedback.

In (1) and (2) subproblems, we will analyze GD with deterministic feedback, where the gradient $\nabla f(\mathbf{x}_t)$ at each point $\mathbf{x}_t$ can be fully observed and GD updates as:

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \eta \nabla f(\mathbf{x}_t). \tag{2.1}$$

(1) **[10pts]** Design an appropriate step size η (L is known), and prove that with the designed step size, GD in (2.1) satisfies:

$$f(\mathbf{x}_{t+1}) \leq f(\mathbf{x}_t) - \frac{1}{2L}\|\nabla f(\mathbf{x}_t)\|_2^2.$$

(2) **[10pts]** Prove that, GD in (2.1) with η designed in subproblem (1) guarantees:

$$\sum_{t=1}^{T} \|\nabla f(\mathbf{x}_t)\|_2^2 \leq \mathcal{O}\left(L\Delta\right),$$

where $\Delta \triangleq f(\mathbf{x}_1) - \min_{\mathbf{x}\in\mathbb{R}^d} f(\mathbf{x})$. Furthermore, let $\tilde{\mathbf{x}}$ be a decision uniformly selected from $\mathbf{x}_1, \ldots, \mathbf{x}_T$, then, with the designed step size, prove that:

$$\mathbb{E}\left[\|\nabla f(\tilde{\mathbf{x}})\|_2\right] \leq \mathcal{O}\left(\frac{\sqrt{L\Delta}}{\sqrt{T}}\right),$$

i.e., the convergence rate to an ε-appropriate point is $\mathcal{O}(1/\sqrt{T})$.

In (3) and (4) subproblems, we will analyze GD under stochastic feedback, where at each round t, the algorithm provides a decision $\mathbf{x}_t$, and only a noisy gradient $\mathbf{g}_t \in \mathbb{R}^d$ can be observed. We assume that the noisy gradient is:

(i) unbiased: $\mathbb{E}\left[\mathbf{g}_t\right] = \nabla f(\mathbf{x}_t)$; (ii) variance-bounded: $\mathbb{E}\left[\|\mathbf{g}_t - \nabla f(\mathbf{x}_t)\|_2^2\right] \leq \sigma^2$.

Additionally, we assume (iii) the evaluations of gradients are independent across iterations. Accordingly, GD updates as:

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \eta \mathbf{g}_t. \tag{2.2}$$

(3) **[15pts]** Prove that GD in (2.2) satisfies:

$$\mathbb{E}[f(\mathbf{x}_{t+1})] \leq \mathbb{E}[f(\mathbf{x}_t)] + \left(\frac{L\eta^2}{2} - \eta\right) \mathbb{E}[\|\nabla f(\mathbf{x}_t)\|_2^2] + \frac{L\eta^2}{2}\sigma^2,$$

where the expectation is taken with respect to the randomness of stochastic gradients.

(**Hint:** The stochastic gradient is unbiased.)

(4) **[15pts]** Prove that when $\eta \leq \frac{1}{L}$, GD in (2.2) satisfies:

$$\mathbb{E}\left[\sum_{t=1}^{T} \|\nabla f(\mathbf{x}_t)\|_2^2\right] \leq \mathcal{O}\left(\frac{\Delta}{\eta} + \eta L T \sigma^2\right).$$

Let $\tilde{\mathbf{x}}$ be a decision uniformly selected from $\mathbf{x}_1, \ldots, \mathbf{x}_T$. Then, try to design a step size η (L, σ, Δ and T are known), and prove that, with the designed step size:

$$\mathbb{E}[\|\nabla f(\tilde{\mathbf{x}})\|_2] \leq \mathcal{O}\left(\frac{\sqrt{L\Delta}}{\sqrt{T}} + \frac{\sqrt{\sigma\sqrt{L\Delta}}}{T^{1/4}}\right),$$

which indicates that the convergence rate to an ε-appropriate point is $\mathcal{O}(1/\sqrt{T} + \sqrt{\sigma}/T^{1/4})$, and when $\sigma = 0$, i.e., there is no randomness, the above result recovers the $\mathcal{O}(1/\sqrt{T})$ convergence rate with deterministic feedback.

(**Hint:** You may need to consider a case-by-case analysis for step size tuning.)

Solution. Give your answers here. (中英文均可)

3 [50pts] OMD with Time-Varying Comparators

In this problem, we are interested in benchmarking the performance of Online Gradient Descent (OGD) against time-varying comparators:

$$\sum_{t=1}^{T} f_t(\mathbf{x}_t) - \sum_{t=1}^{T} f_t(\mathbf{u}_t),$$

where $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_T \in \mathcal{X}$ are arbitrary comparators in the feasible domain. By choosing $\mathbf{u}_1 = \cdots = \mathbf{u}_T = \mathbf{x}_\star$, where $\mathbf{x}_\star \in \arg\min_{\mathbf{x}\in\mathcal{X}} \sum_{t=1}^{T} f_t(\mathbf{x})$, this measure recovers the standard regret discussed in the class. While the flexibility in choosing $\mathbf{u}_1, \ldots, \mathbf{u}_T$ allows for the algorithm to handle more complex settings.

During the subsequent subproblems, our analysis will be centered on OGD:

$$\mathbf{x}_{t+1} = \Pi_{\mathcal{X}} \left[\mathbf{x}_t - \eta \nabla f_t(\mathbf{x}_t) \right]. \tag{3.1}$$

We assume that the domain diameter is bounded by D, i.e., $\sup_{\mathbf{x},\mathbf{y}} \|\mathbf{x} - \mathbf{y}\|_2 \le D$, and the gradient norm is bounded by G, i.e., $\|\nabla f_t(\mathbf{x})\|_2 \le G, \forall t \in [T], \mathbf{x} \in \mathcal{X}$. For simplicity, we assume $f_t(\mathbf{x}) \in [0, GD], \forall \mathbf{x} \in \mathcal{X}, t \in [T]$.

(1) **[5pts]** Try to prove the following property:

$$\|\mathbf{x} - \mathbf{y}\|_2^2 - \|\mathbf{x} - \mathbf{z}\|_2^2 \le 4D\|\mathbf{y} - \mathbf{z}\|_2, \forall \mathbf{x}, \mathbf{y}, \mathbf{z} \in \mathcal{X}.$$

(2) **[10pts]** Try to prove that OGD in (3.1) satisfies the following regret bound:

$$\sum_{t=1}^{T} f_t(\mathbf{x}_t) - \sum_{t=1}^{T} f_t(\mathbf{u}_t) \le \frac{4DQ_T + D^2}{2\eta} + \frac{\eta G^2 T}{2}, \tag{3.2}$$

where we introduce $Q_T = \sum_{t=2}^{T} \|\mathbf{u}_t - \mathbf{u}_{t-1}\|_2$ in above, a quantity measuring the changing degree in the comparators.

(3) **[20pts]** If we could know the exact value of Q_T in advance, by setting $\eta = \mathcal{O}\big(\sqrt{\frac{1+Q_T}{T}}\big)$, we can obtain the regret bound of $\mathcal{O}(\sqrt{(1+Q_T)T})$. However, as we expect our method to hold consistently for any $\mathbf{u}_1, \ldots, \mathbf{u}_T$, assuming the knowledge of Q_T is unrealistic.

Instead of tuning a single algorithm, we can estimate the value of Q_T and run multiple instances of the OGD algorithm to offset the uncertainty. At last, we will combine the decisions from different instances via Hedge. We describe this method in below:

$$\mathcal{H} = \left\{ \eta_i = 2^{i-1} \cdot \frac{D}{G\sqrt{T}} : i \in [N] \right\} \tag{3.3}$$

$$\mathbf{x}_{t+1,i} = \arg\min_{\mathbf{x}\in\mathcal{X}} \left\{ \langle \nabla f_t(\mathbf{x}_{t,i}), \mathbf{x} \rangle + \frac{1}{2\eta_i} \|\mathbf{x} - \mathbf{x}_{t,i}\|_2^2 \right\}, \qquad \forall i \in [N] \tag{3.4}$$

$$p_{t+1,i} \propto \exp\Big(-\varepsilon \sum_{s=1}^{t} f_s(\mathbf{x}_{s,i})/GD \Big), \boldsymbol{p}_1 = \frac{1}{N} \cdot \mathbf{1} \qquad \forall i \in [N]. \tag{3.5}$$

$$\mathbf{x}_{t+1} = \sum_{i=1}^{N} p_{t+1,i}\mathbf{x}_{t+1,i} \tag{3.6}$$

In above, $N = \lceil \frac{1}{2}\log_2(1+4T) \rceil$ denotes the number of running OGD instances. (3.3) is the possible step sizes, and for each $\eta_i \in \mathcal{H}$, we employ an OGD with the specific step size η_i, as presented in (3.4). Eq. (3.5) calculates the weights for combining via Hedge taught in the class. Finally, in Eq (3.6), $\mathbf{x}_{t+1}$ is the final decision we submit.

(3.i) **[5pts]** The ideal step size for Eq. (3.2) is:

$$\eta_\star = \sqrt{\frac{4DQ_T + D^2}{G^2 T}}.$$

Prove that given any arbitrary comparators $\mathbf{u}_1, \cdots, \mathbf{u}_T$, there exists $\eta_{i_\star} \in \mathcal{H}$, such that the following inequality holds:

$$\eta_{i_\star} \leq \eta_\star \leq 2\eta_{i_\star}.$$

(3.ii) **[5pts]** Design the learning rate ε in (3.5) and prove that:

$$\sum_{t=1}^{T} f_t(\mathbf{x}_t) - \sum_{t=1}^{T} f_t(\mathbf{x}_{t,i}) \leq \mathcal{O}\left(\sqrt{T}\right), \forall i \in [N],$$

where in above, we treat doubly-logarithmic factor $\mathcal{O}(\log\log T)$ as a constant.

(3.iii) **[10pts]** Prove that, with the learning rate ε satisfying the requirement in problem (3.ii), decisions $\{\mathbf{x}_t\}_{t=1}^T$ generated by (3.6) guarantee:

$$\sum_{t=1}^{T} f_t(\mathbf{x}_t) - \sum_{t=1}^{T} f_t(\mathbf{u}_t) \leq \mathcal{O}\left(\sqrt{(1+Q_T)T}\right),$$

for any arbitrary comparators $\mathbf{u}_1, \ldots, \mathbf{u}_T \in \mathcal{X}$.

(4) **[15pts]** Method described from (3.3) to (3.6) requires multiple queries of gradients ($\nabla f_t(\mathbf{x}_{t,i})$) and function values ($f_t(\mathbf{x}_{t,i})$) at each time. Can you improve this method and develop a more efficient one that only queries one gradient $\nabla f_t(\mathbf{x}_t)$ at each time? Present your method in a format similar to Eq. (3.3) to Eq. (3.6), specifying the corresponding step sizes and learning rate. Highlight how you will analyze the regret from the combination and the regret for the running instance.

Solution. Give your answers here. (中英文均可)

4 [50pts] Learning Rate Tuning in (Adaptive) Hedge

Consider the Prediction with Experts' Advice (PEA) problem, where we denote $\boldsymbol{\ell}_t \in [0,1]^N$ to be the loss vector at time $t \in [T]$, and the domain is the simplex Δ_N. One of the classic PEA algorithms is Hedge, which updates the weights as follows,

$$p_{t+1,i} \propto \exp(-\eta L_{t,i}), \forall i \in [N], \tag{4.1}$$

where $L_{t,i} = \sum_{s=1}^{t} \ell_{s,i}$ is the cumulative loss of the i-th expert.

(1) **[10pts]** Prove that the Hedge algorithm with the learning rate η ensures that:

$$\sum_{t=1}^{T} \langle \boldsymbol{p}_t, \boldsymbol{\ell}_t \rangle - L_{T,i^\star} \leq \frac{\ln N}{\eta} + \eta \sum_{t=1}^{T} \langle \boldsymbol{p}_t, \boldsymbol{\ell}_t \rangle,$$

where $L_{T,i^\star} = \min_{i \in [N]} L_{T,i}$. Then further prove that the regret can be bounded by $\mathcal{O}(\sqrt{T \log N})$ with the optimal tuning η when T is given.

(2) **[15pts]** The Hedge algorithm achieves a regret bound of $\mathcal{O}(\sqrt{T \log N})$ with optimal tuning of η, provided that T is known in advance. However, what if the total iterations T are unknown? One of the approaches to address it is to employ time-varying learning rates. However, this approach requires a new analysis and redesign of the algorithm itself. In the following, we instead aim to develop a tuning strategy that leverages the results studied so far in a black-box manner to overcome it, without the need for the time-varying learning rates design.

The approach is to start with an initial guess for T, and whenever the actual number of iterations exceeds this guess, we double the guess and restart the algorithm. The main idea is summarized in Algorithm 1 (with $\mathbf{0}$ being the all-zero vector). Two blanks, (i) and (ii), remain for you to fill in. Then try to prove that Algorithm 1 ensures $\mathcal{O}(\sqrt{T \log N})$ for all T.

(**Hint**: Consider the regret between two resets and take the summation of them.)

Algorithm 1 Hedge with Black-box Tuning

Initialization: Set $L_0 = \mathbf{0}, T_0 = 1, \eta = \sqrt{(\ln N)/T_0}$.
for $t = 1, 2, \ldots$ **do**
  Compute $\boldsymbol{p}_t$ by (4.1)
  Play $\boldsymbol{p}_t$ and receive $\boldsymbol{\ell}_t$
  $L_t = L_{t-1} + \boldsymbol{\ell}_t$
  if $t = T_0$ **then**
    $L_t = \mathbf{0}, \eta =$ (i)
    $T_0 \leftarrow$ (ii)
  end if
end for

(3) **[15pts]** Beyond achieving the regret bound of $\mathcal{O}(\sqrt{T \log N})$, we are interested in obtaining a more adaptive bound that replaces the dependence of T by $L_{T,i_\star}$. This type of bound in $\mathcal{O}(\sqrt{L_{T,i_\star} \log N})$ is known as "small-loss" bound, where the algorithm's performanc scales with the cumulative loss of the best expert $i_\star$.

Prove that Hedge with fixed learning rate η ensures that

$$\sum_{t=1}^{T} \langle \boldsymbol{p}_t, \boldsymbol{\ell}_t \rangle - L_{T,i^\star} \leq \frac{1}{1-\eta} \left(\frac{\ln N}{\eta} + \eta L_{T,i^\star} \right),$$

where the tuning $\eta = \min\{\frac{1}{2}, \sqrt{(\ln N)/L_{T,i^\star}}\}$ achieves $\mathcal{O}(\sqrt{L_{T,i^\star} \log N} + \log N)$. However, the quantity $L_{T,i^\star}$ is unknown in advance; nonetheless, one can still use the same tuning idea presented previously to achieve the same bound.

Try to design a tuning strategy similar to the spirit of subproblem (2), such that the bound $\mathcal{O}(\sqrt{L_{T,i^\star} \log N} + \log N)$ can be obtained without knowing $L_{T,i^\star}, T$ and $i^\star$ in advance.

(4) **[10pts]** Try to prove the regret bound $\mathcal{O}(\sqrt{L_{T,i^\star} \log N} + \log N)$ of the method you have designed in subproblem (3).

Solution. Give your answers here. (中英文均可)

5 [50pts] OMD with a Stabilizer

The classic Online Mirror Descent (OMD) algorithm follows the below update formula:

$$\mathbf{x}_{t+1} = \arg\min_{\mathbf{x}\in\mathcal{X}} \left\{\eta_t \langle \nabla f_t(\mathbf{x}_t), \mathbf{x}\rangle + \mathcal{D}_\psi(\mathbf{x}, \mathbf{x}_t)\right\}. \tag{5.1}$$

A similar online algorithm, Follow the Regularized Leader (FTRL), updates as:

$$\mathbf{x}_{t+1} = \arg\min_{\mathbf{x}\in\mathcal{X}} \left\{\eta_t \sum_{s=1}^{t} \langle \nabla f_s(\mathbf{x}_s), \mathbf{x}\rangle + \psi(\mathbf{x})\right\}. \tag{5.2}$$

During the course, we studied that OMD and FTRL are equivalent under certain conditions when employing the same fixed step size. However, in general, they are different particularly when the step size can change over time, and we now investigate the difference.

(1) **[10pts]** We set the regularizer $\psi(\mathbf{x}) = \frac{1}{2}\|\mathbf{x} - \mathbf{x}_1\|_2^2$, and corresponding step sizes $\eta_{t+1} \le \eta_t, \forall t \in [T]$ for OMD and FTRL in (5.1) and (5.2). Under these conditions, try to prove that OMD presented in (5.1) guarantees that:

$$\sum_{t=1}^{T} f_t(\mathbf{x}_t) - \sum_{t=1}^{T} f_t(\mathbf{x}_\star) \le \mathcal{O}\left(\frac{\max_{t\in[T]}\|\mathbf{x}_t - \mathbf{x}_\star\|_2^2}{\eta_T} + \sum_{t=1}^{T} \eta_t \|\nabla f_t(\mathbf{x}_t)\|_2^2\right),$$

where $\mathbf{x}_\star \in \arg\min_{\mathbf{x}\in\mathcal{X}} \sum_{t=1}^{T} f_t(\mathbf{x})$.

Additionally, prove the following regret bound for FTRL presented in (5.2):

$$\sum_{t=1}^{T} f_t(\mathbf{x}_t) - \sum_{t=1}^{T} f_t(\mathbf{x}_\star) \le \mathcal{O}\left(\frac{\|\mathbf{x}_1 - \mathbf{x}_\star\|_2^2}{\eta_T} + \sum_{t=1}^{T} \eta_{t-1} \|\nabla f_t(\mathbf{x}_t)\|_2^2\right).$$

(2) **[10pts]** In subproblem (1), we notice that the regret bound for OMD depends on the factor $\max_{t\in[T]}\|\mathbf{x}_t - \mathbf{x}_\star\|_2^2$, which is challenging to analyze further and could be arbitrarily large in some cases. While for FTRL, the factor $\|\mathbf{x}_1 - \mathbf{x}_\star\|_2^2$ shown in the bound is irrelevant to the decision process, and could be small if we choose a good starting point with prior knowledge. Next, we consider a specific setting where this point could lead to significantly different results.

Consider the Prediction with Experts' Advice (PEA) setting, where the domain is $\mathcal{X} = \Delta_N$. We often choose ψ as the negative entropy function, and in this case the induced Bregman divergence $\mathcal{D}_\psi(\cdot,\cdot)$ becomes the well-known KL-divergence. We set the starting point as $\mathbf{x}_1 = [1/N, \ldots, 1/N]$. With this setup, try to prove that:

$$\sup_{\mathbf{x}\in\mathcal{X}} \mathcal{D}_\psi(\mathbf{x}, \mathbf{x}_1) \le \ln N, \qquad \sup_{\mathbf{x},\mathbf{y}\in\mathcal{X}} \mathcal{D}_\psi(\mathbf{x}, \mathbf{y}) = +\infty.$$

(3) **[15pts]** We expect that OMD can exhibit similarly desirable properties as FTRL. For this purpose, we consider the following modified OMD under a simpler OCO setting:

$$\mathbf{x}_{t+1} = \arg\min_{\mathbf{x}\in\mathcal{X}} \left\{ \eta_t \langle \nabla f_t(\mathbf{x}_t), \mathbf{x} \rangle + \frac{1}{2}\|\mathbf{x} - \mathbf{x}_t\|_2^2 + \Big(\frac{\eta_t}{\eta_{t+1}} - 1\Big)\|\mathbf{x} - \mathbf{x}_1\|_2^2 \right\}.$$

In above, we introduce a stabilizer (the last term in above) to the update formula. In a sense, if $\eta_t = \eta_{t+1}$, which recovers to the fixed step size setting, this regularizer becomes zero. However, if the step sizes decrease too rapidly, then $\frac{\eta_t}{\eta_{t+1}} - 1 > 0$ will become larger and the stabilizer will "drag" the decision closer to $\mathbf{x}_1$.

Try to prove the following inequality:

$$\begin{aligned}\langle \nabla f_t(\mathbf{x}_t), \mathbf{x}_t - \mathbf{x}_\star \rangle \leq \frac{1}{2\eta_t}\left(\|\mathbf{x}_t - \mathbf{x}_\star\|_2^2 - \|\mathbf{x}_{t+1} - \mathbf{x}_\star\|_2^2 - \|\mathbf{x}_t - \mathbf{x}_{t+1}\|_2^2\right) \\ + \langle \nabla f_t(\mathbf{x}_t), \mathbf{x}_t - \mathbf{x}_{t+1} \rangle + \phi(\mathbf{x}_\star) - \phi(\mathbf{x}_{t+1}),\end{aligned}$$

where we define $\phi(\mathbf{x}) = (\frac{1}{\eta_{t+1}} - \frac{1}{\eta_t})\|\mathbf{x} - \mathbf{x}_1\|_2^2$.

(**Hint:** $\phi(\mathbf{x})$ is convex.)

(4) **[15pts]** Assume that $\eta_{t+1} \leq \eta_t, \forall t \in [T]$, prove that the following regret bound for OMD with stabilizer:

$$\sum_{t=1}^{T} f_t(\mathbf{x}_t) - \sum_{t=1}^{T} f_t(\mathbf{u}) \leq \mathcal{O}\left(\frac{\|\mathbf{x}_1 - \mathbf{x}_\star\|_2^2}{\eta_{T+1}} + \sum_{t=1}^{T} \eta_t \|\nabla f_t(\mathbf{x}_t)\|_2^2\right).$$

(**Hint:** Think about which terms contribute to $\max_{t\in[T]}\|\mathbf{x}_t - \mathbf{x}_\star\|_2^2$.)

Solution. Give your answers here. (中英文均可)

6 [5pts] Bonus (Lecture Slides 1-7)

You can earn bonus points by pointing out errors in the lecture slides 1-7 on the course website. Specifically, consider the following three types of errors:

(A) Technical errors (e.g., incorrect coefficients in formulas), 1pts each.

(B) Serious typo in presentation (e.g., AB but actually $A^\top B$, $\mathbf{x}A$ but actually $\mathbf{x}^\top A$), 0.5pts each.

(C) Typos in formula/statement (e.g., writing vector $\mathbf{x}_t$ as x_t; grammar/spelling errors), 0.25pts each.

(D) Other suggestions: like how to better organize the proofs or alternative simplified proofs..., up to 1.5pts each.

List the errors in lecture slides 1-7 and state the way to correct. Please clearly indicate which type each error belongs to, with a total score not exceeding 5pts.

For example,

(1) [**(A) Technical errors**] Lecture X. Page2. xxx

(2) [**(B) Serious typo in presentation**] Lecture Y. Page4. yyy → zzz

(3) [**(C) Typos in formula/statement**] Lecture W. Page6. www → vvv

(4) [**(D) Other suggestions**] Lecture V. Page8. It would be better...

Solution. Give your answers here. (中英文均可)

Acknowledgements

The homework bearing your name must represent your individual contribution. While discussions during the completion of the assignment are permissible, they are conditioned upon the fact that none of the participating individuals have completed the discussed topics. We emphasize that the implementation of key ideas within the assignment must be done independently. **You should extend your acknowledgments to those individuals who have participated in the discussions here**.

This course adopts a zero-tolerance policy toward plagiarism. The grades of students found to have engaged in plagiarism without providing proper citations or acknowledgments will be **annulled**. In cases of mutual plagiarism, the grades of **both** the plagiarizer and the plagiarized will be **annulled**.