ONLINE OPTIMIZATION HW-2

SIDDHANT CHAUDHARY
BMC201953

Problem 1. Give an algorithm to project a point $\boldsymbol{x} \in \mathbf{R}^n$ to the n-simplex, $\sum_i x_i = 1$, $1 \geq x_i \geq 0$ for all i.

Solution. For the pseudocode, please refer to **Algorithm** 1.

We will now give a description of our algorithm. Consider the n-simplex $\Delta_n \subset \mathbf{R}^n$. We know that the vertices of this simplex are the standard basis vectors $\{\boldsymbol{e}_1, ..., \boldsymbol{e}_n\}$, and Δ_n is nothing but the convex hull of this set. Now, note that the convex hull of any subset $S \subset \{\boldsymbol{e}_1, ..., \boldsymbol{e}_n\}$ is a *facet* of the n-simplex.

Our algorithm consists of a function PROJECT which takes as arguments a point $\boldsymbol{x} \in \mathbf{R}^n$ and a set $S \subseteq \{\boldsymbol{e}_1, ..., \boldsymbol{e}_n\}$; it returns a pair $(\boldsymbol{p}, \boldsymbol{d})$, where $\boldsymbol{p}$ is the projection of $\boldsymbol{x}$ onto the convex hull of S, and $d = ||\boldsymbol{x} - \boldsymbol{p}||$, i.e the distance between the point and the projection. So, the final answer will be PROJECT$(\boldsymbol{x}, \{\boldsymbol{e}_1, ..., \boldsymbol{e}_n\})$.

Let us now describe the algorithm.

(1) Suppose our input is $\boldsymbol{x} \in \mathbf{R}^n$, and $S \subseteq \{\boldsymbol{e}_1, ..., \boldsymbol{e}_n\}$ is some subset. Suppose $\boldsymbol{S} = \{\boldsymbol{v}_1, ..., \boldsymbol{v}_k\}$.
(2) If $k = 1$, then the returned value must be $(\boldsymbol{v}_1, ||\boldsymbol{x} - \boldsymbol{v}_1||)$, and that is what our algorithm does.
(3) So suppose $k > 1$. Note that the convex hull of points in S lies in a *translated* $k-1$-dimensional vector space. For example, in $\mathbf{R}^3$, if we take $S = \{\boldsymbol{e}_1, \boldsymbol{e}_3\}$, then their convex hull, which is just the line segment between $\boldsymbol{e}_1, \mathbf{e}_3$, is really a subset of the line containing that line segment. This line can be thought of as a translation of a 1-dimensional vector subspace of $\mathbf{R}^3$. So, lines 12-14 of the algorithm shifts the origin to a point in S (specifically, point $\boldsymbol{v}_1$), so that the convex hull lies in an actual vector subspace. This is done because working with vector subspaces is easier than working with their translations.
(4) It is easy to see that the vectors $\{\boldsymbol{v}_2 - \boldsymbol{v}_1, ..., \boldsymbol{v}_k - \boldsymbol{v}_1\}$ are actually linearly independent (it is easy to see this because these are standard basis vectors). So, they span the $k-1$ dimensional vector space containing them. Line 18 just converts this basis to an orthonormal basis using the usual Gram-Schmidt technique. Suppose the orthonormal basis obtained is $\{\boldsymbol{v}_2', ..., \boldsymbol{v}_k'\}$.
(5) Lines 20-21 find the coordinates of the projection of the point $\boldsymbol{x} - \boldsymbol{v}_1$ (the translated point) onto this $k-1$-dimensional space w.r.t the basis $\{\boldsymbol{v}_2', ..., \boldsymbol{v}_k'\}$. Here we are just using the fact that the vector between the point $\boldsymbol{x} - \boldsymbol{v}_1$ and the projection is orthogonal to the space spanned by $\{\boldsymbol{v}_2', ..., \boldsymbol{v}_k'\}$; hence the formula (in the algorithm) for the c_is holds. This projection is called $\boldsymbol{x}'$.
(6) Then, it is checked if $\boldsymbol{x}' + \boldsymbol{v}_1$ (note that we are adding $\boldsymbol{v}_1$ back to go back to the original coordinate system) is contained in the convex hull of the points in

Date: 4 November, 2021.

Algorithm 1 Algorithm to project onto n-simplex Δ_n

1: **Input**: A point $\boldsymbol{x} \in \mathbf{R}^n$.
2:
3: **function** PROJECT($\boldsymbol{x}, S$) ▷ S is a subset of the standard basis $\{\boldsymbol{e}_1, ..., \boldsymbol{e}_n\}$
4: Let C = convex hull of S.
5: The function will return the pair $(\Pi_C(\boldsymbol{x}), ||\boldsymbol{x} - \Pi_C(\boldsymbol{x})||)$.
6:
7: Suppose $S = \{\boldsymbol{v}_1, ..., \boldsymbol{v}_k\} \subseteq \{\boldsymbol{e}_1, ..., \boldsymbol{e}_n\}$.
8: **if** $k = 1$ **then** ▷ Handling the boundary case
9: **return** $(\boldsymbol{v}_1, ||\boldsymbol{x} - \boldsymbol{v}_1||)$
10: **end if**
11:
12: $\boldsymbol{x} \leftarrow \boldsymbol{x} - \boldsymbol{v}_1$ ▷ Lines 12, 13, 14 shift the origin to $\boldsymbol{v}_1$
13: **for** $i = 2$ to k **do**
14: $\boldsymbol{v}_i \leftarrow \boldsymbol{v}_i - \boldsymbol{v}_1$
15: **end for**
16:
17: Now, $\{\boldsymbol{v}_2, ..., \boldsymbol{v}_k\}$ is a basis of $\text{span}(\boldsymbol{v}_2, ..., \boldsymbol{v}_k)$.
18: Use Gram-Schmidt to convert $\{\boldsymbol{v}_2, ..., \boldsymbol{v}_k\}$ to an orthonormal basis $\{\boldsymbol{v}'_2, ..., \boldsymbol{v}'_k\}$.
19:
20: **for** $i = 2$ to k **do**
21: $c_i \leftarrow \langle \boldsymbol{x}, \boldsymbol{v}'_i \rangle$ ▷ $\sum_{i=2}^{k} c_i \boldsymbol{v}'_i$ is the projection of $\boldsymbol{x}$ onto $\text{span}(\boldsymbol{v}_2, ..., \boldsymbol{v}_k)$
22: **end for**
23:
24: $\boldsymbol{x}' \leftarrow \sum_{i=2}^{k} c_i \boldsymbol{v}'_i$
25: $d_0 \leftarrow ||\boldsymbol{x} - \boldsymbol{x}'||$
26: **if** $\boldsymbol{x}' + \boldsymbol{v}_1 \in C$ **then** ▷ This can be checked easily
27: **return** $(\boldsymbol{x}' + \boldsymbol{v}_1, d_0)$
28: **end if**
29:
30: $\boldsymbol{x}' \leftarrow \boldsymbol{x}' + \boldsymbol{v_1}$
31: **for** $i = 2$ to k **do** ▷ Lines 30, 31, 32 shift the origin back to 0
32: $\boldsymbol{v}_i \leftarrow \boldsymbol{v}_i + \boldsymbol{v}_1$
33: **end for**
34:
35: $(\boldsymbol{p}, d) \leftarrow (\mathbf{0}, \infty)$ ▷ Initialise the pair to be returned
36: **for** $i = 1$ to k **do**
37: $S' \leftarrow S - \{\boldsymbol{v}_i\}$
38: $(\boldsymbol{p}', d') \leftarrow \text{PROJECT}(\boldsymbol{x}', S')$
39: **if** $d' < d$ **then**
40: $(\boldsymbol{p}, d) \leftarrow (\boldsymbol{p}', d')$
41: **end if**
42: **end for**
43: **return** $(\boldsymbol{p}, \sqrt{d^2 + d_0^2})$ ▷ By **Pythagoras Theorem**
44: **end function**
45:
46: **Output**: PROJECT($\boldsymbol{x}, \{\boldsymbol{e}_1, ..., \boldsymbol{e}_n\}$)

S; this is easy to check, as it can be checked by verifying whether the equality

$$\sum_{i=1}^{k} \langle \boldsymbol{x}' + \boldsymbol{v}_1, \boldsymbol{v}_i \rangle = 1$$

holds (note that we are using the fact that $\{\boldsymbol{v}_1, ..., \boldsymbol{v}_k\} \subseteq \{\boldsymbol{e}_1, ..., \boldsymbol{e}_n\}$), and hence checking inclusion in convex hull is easy. If yes, the projection is simply $\boldsymbol{x}' + \boldsymbol{v}_1$, and along that we return the distance between $\boldsymbol{x} - \boldsymbol{v}_1$ and $\boldsymbol{x}'$.

(7) So suppose the answer to the previous point is not. Then first, lines 30-32 shift the coordinate system back to the origin by adding $\boldsymbol{v}_1$ to each vector. So, we are back in our original space, and the vector $\boldsymbol{x}' + \boldsymbol{v}_1$ is the projection of the point $\boldsymbol{x}$ to the $k-1$-dimensional translated vector space containing the convex hull of S.

(8) Now, note that the projection of $\boldsymbol{x}$ onto the convex hull of S is nothing but the projection of $\boldsymbol{x}' + \boldsymbol{v}_1$ onto the convex hull (**Pythagoras Theorem**); since $\boldsymbol{x}' + v_1$ and the convex hull all lie in the same $k-1$-dimensional vector space, we have reduced the dimension of the problem by 1, and we can hence solve it recursively.

(9) Now, since the point $\boldsymbol{x}' + \boldsymbol{v}_1$ lies outside the convex hull, it's projection will be on the boundary of the convex hull, i.e it will be on some *facet* of the convex hull. Any boundary facet will be the convex hull of any $k-1$-sized subset of S. So, for each $k-1$-sized subset S' of S, we recursively compute the distance between $\boldsymbol{x}' + \boldsymbol{v}_1$ and it's projection onto the convex hull of points in S'; the least among these distances will be the actual distance between the point $\boldsymbol{x}' + \boldsymbol{v}_1$ and it's projection onto the convex hull of points in S.

(10) We simply return the distance $\sqrt{d^2 + d_0^2}$ as our answer; this is nothing but the **Pythagoras Theorem**, and we return the point $\boldsymbol{p}$ which realizes this distance from $\boldsymbol{x}$. Hence, the point $\boldsymbol{p}$ will be the required projection.

I haven't checked the time complexity of this algorithm, but it looks like poly(n). ■

Problem 2. Assume access to Nesterov's algorithm that attains a rate of $e^{-\sqrt{\gamma}T}$ for a γ-well conditioned function. Apply a reduction to obtain a β/T^2 rate for β-smooth functions (upto log factors).

Solution. As usual, let $\mathcal{K}$ be a convex body, and let $f : \mathcal{K} \to \mathbf{R}$ be a β-smooth differentiable function. Let $\boldsymbol{x}^*$ be the minimizer of f over $\mathcal{K}$. Let D be the diameter of the convex set $\mathcal{K}$.

Since we have access to Nestorov's algorithm which only works for γ-well conditioned functions, we do the following: suppose the initial point to be fed to the algorithm is $\boldsymbol{x}_1 \in \mathcal{K}$. Define the function $g : \mathcal{K} \to \mathbf{R}$ as follows.

$$g(\boldsymbol{x}) = f(\boldsymbol{x}) + \frac{\alpha}{2} ||\boldsymbol{x} - \boldsymbol{x}_1||^2$$

Above, α is some number which will be determined in a moment. Observe that the function $h(\boldsymbol{x}) = \frac{\alpha}{2} ||\boldsymbol{x} - \boldsymbol{x}_1||^2$ is both α-strongly convex and α-smooth; this is true because for any $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{K}$, the following equation is true.

$$\begin{aligned} h(\boldsymbol{x}) = \frac{\alpha}{2}\left|\left|\boldsymbol{x} - \boldsymbol{x}_1\right|\right|^2 &= \frac{\alpha}{2}\left|\left|\boldsymbol{y} - \boldsymbol{x}_1\right|\right|^2 + \frac{\alpha}{2}\left|\left|\boldsymbol{x} - \boldsymbol{y}\right|\right|^2 + 2\frac{\alpha}{2}\left\langle \boldsymbol{y} - \boldsymbol{x}_1, \boldsymbol{x} - \boldsymbol{y}\right\rangle \\ &= \frac{\alpha}{2}\left|\left|\boldsymbol{y} - \boldsymbol{x}_1\right|\right|^2 + \left\langle \nabla h(\boldsymbol{y}), \boldsymbol{x} - \boldsymbol{y}\right\rangle + \frac{\alpha}{2}\left|\left|\boldsymbol{x} - \boldsymbol{y}\right|\right|^2 \\ &= h(\boldsymbol{y}) + \left\langle \nabla h(\boldsymbol{y}), \boldsymbol{x} - \boldsymbol{y}\right\rangle + \frac{\alpha}{2}\left|\left|\boldsymbol{x} - \boldsymbol{y}\right|\right|^2 \end{aligned}$$

So, it follows that the function g is α-strongly convex and $\alpha + \beta$-smooth; i.e, the function g is $\frac{\alpha}{\alpha+\beta}$-well conditioned. So, let

$$\gamma = \frac{\alpha}{\alpha + \beta}$$

Now, let $h_t = f(\boldsymbol{x}) - f(\boldsymbol{x}^*)$ and let $h_t^g = g(\boldsymbol{x}_t) - g(\boldsymbol{x}_g^*)$, where $\boldsymbol{x}_g^* \in \mathcal{K}$ is the minimizer of g. Clearly, we have that $g(\boldsymbol{x}^*) \geq g(\boldsymbol{x}_g^*)$.

We run Nesterov's algorithm with initial point $\boldsymbol{x}_1$ on the function g. Now, observe the following.

$$\begin{aligned} h_t &= g(\boldsymbol{x}_t) - g(\boldsymbol{x}^*) + \frac{\alpha}{2}\left|\left|\boldsymbol{x}^* - \boldsymbol{x}_1\right|\right|^2 - \frac{\alpha}{2}\left|\left|\boldsymbol{x}_t - \boldsymbol{x}_1\right|\right|^2 \\ &\leq g(\boldsymbol{x}_t) - g(\boldsymbol{x}_g^*) + \alpha D^2 \\ &= h_t^g + \alpha D^2 \end{aligned}$$

By the convergence guarantee of Nesterov's algorithm, we have the following using the above inequality.

$$\begin{aligned} h_t &\leq h_t^g + \alpha D^2 \\ &\leq h_1^g e^{-\sqrt{\gamma} t} + \alpha D^2 \end{aligned}$$

Now, we will choose

$$\alpha = \frac{\beta \log t}{D^2 t^2}$$

This gives us the following.

$$\gamma = \frac{\alpha}{\alpha + \beta} = \frac{\log t}{\log t + D^2 t^2}$$

For large t, we know that

$$\log t \leq D^2 t^2$$

This means, for large t, we have

$$\frac{\log t}{\log t + D^2 t^2} \geq \frac{\log t}{2 D^2 t^2} \geq \frac{1}{2 D^2 t^2}$$

The above inequality implies that for large t,

$$e^{\sqrt{\gamma} t} \geq e^{\sqrt{\frac{1}{2D^2 t^2}} t} = e^{\sqrt{\frac{1}{2D^2}}}$$

which implies that

$$e^{-\sqrt{\gamma} t} \leq e^{-\sqrt{\frac{1}{2D^2}}}$$

for large t, which implies that $e^{-\sqrt{\gamma} t} = O(1)$. So, this means that

$$h_1^g e^{-\sqrt{\gamma} t} + \alpha D^2 = O\left(\frac{\beta \log t}{t^2}\right)$$

where above we are ignoring the constant h_1^g (which is positive). So, we have shown that with the choice $\alpha = \frac{\beta \text{logt}}{t^2}$, we have

$$h_t \leq O\left(\frac{\beta \log t}{t^2}\right)$$

which is what we wanted to show. ■

Problem 3. Show that SGD for a strongly convex function with appropriately chosen η_t converges at $\tilde{O}(1/T)$. You may assume that gradients are bounded by G. Recall that the $\tilde{O}$-notation hides all kinds of log-factors.

Solution. In class, we have proven the following theorem: *Let $\mathcal{K}$ be a convex set, $\boldsymbol{x}_1 \in \mathcal{K}$ an initial point, and T a time horizon. Let f_t be the revealed cost functions. Suppose each f_t is α-strongly convex. Then, doing OGD with step sizes $\eta_t = \frac{1}{\alpha t}$ gives the following regret bound.*

$$\text{regret}_T \leq \frac{G^2}{2\alpha}(1 + \text{log}T)$$

Here G is an upper bound on the gradients. Using this theorem, we will prove the statement given in the problem.

So, let f be an α-strongly convex function. For each t, we define the following function.

$$g_t(\boldsymbol{x}) = \left\langle \tilde{\nabla}_t, \boldsymbol{x} \right\rangle + \frac{\alpha}{2} ||\boldsymbol{x} - \boldsymbol{x}_1||^2$$

Here $\tilde{\nabla}_t$ is the gradient oracle, i.e

$$\tilde{\nabla}_t = \mathcal{O}(\boldsymbol{x}_t)$$

It is clear that g_t is an α-strongly convex function for each t. Next, we have the following.

$$\begin{aligned}
&\mathbf{E}\left[f(\overline{\boldsymbol{x}_T})\right] - f(\boldsymbol{x}^*) \\
&\leq \frac{1}{T}\mathbf{E}\left[\sum_t f(\boldsymbol{x}_t)\right] - f(\boldsymbol{x}^*) && \text{(Jensen's Inequality)} \\
&= \frac{1}{T}\mathbf{E}\left[\sum_t [f(\boldsymbol{x}_t) - f(\boldsymbol{x}^*)]\right] && \text{(Expectation of a constant)} \\
&\leq \frac{1}{T}\mathbf{E}\left[\sum_t \langle \nabla f(\boldsymbol{x}_t), \boldsymbol{x}_t - \boldsymbol{x}^* \rangle - \frac{\alpha}{2} ||\boldsymbol{x}_t - \boldsymbol{x}^*||^2\right] && \text{(Strong convexity)} \\
&= \frac{1}{T}\mathbf{E}\left[\sum_t \left\langle \tilde{\nabla}_t, \boldsymbol{x}_t - \boldsymbol{x}^* \right\rangle - \frac{\alpha}{2} ||\boldsymbol{x}_t - \boldsymbol{x}^*||^2\right] && \text{(Gradient Oracle)}
\end{aligned}$$

Now, using the trivial inequality

$$-\frac{\alpha}{2} ||\boldsymbol{x}_t - \boldsymbol{x}^*||^2 \leq \frac{\alpha}{2} ||\boldsymbol{x}_t - \boldsymbol{x}_1||^2 - \frac{\alpha}{2} ||\boldsymbol{x}^* - \boldsymbol{x}_1||^2$$

we get the following.

$$\frac{1}{T}\mathbf{E}\left[\sum_t \left\langle \tilde{\nabla}_t, \boldsymbol{x}_t - \boldsymbol{x}^*\right\rangle - \frac{\alpha}{2}\left|\left|\boldsymbol{x}_t - \boldsymbol{x}^*\right|\right|^2\right]$$
$$\leq \frac{1}{T}\mathbf{E}\left[\sum_t \left\langle \tilde{\nabla}_t, \boldsymbol{x}_t - \boldsymbol{x}^*\right\rangle + \frac{\alpha}{2}\left|\left|\boldsymbol{x}_t - \boldsymbol{x}_1\right|\right|^2 - \frac{\alpha}{2}\left|\left|\boldsymbol{x}^* - \boldsymbol{x}_1\right|\right|^2\right]$$
$$= \frac{1}{T}\mathbf{E}\left[\sum_t g_t(\boldsymbol{x}_t) - g_t(\boldsymbol{x}^*)\right] \qquad \text{(Definition of } g_t\text{)}$$
$$\leq \frac{\text{regret}_T}{T}$$
$$\leq \frac{G^2}{2\alpha}\frac{(1+\log T)}{T} \qquad \text{(By theorem mentioned above)}$$
$$= \tilde{O}\left(\frac{1}{T}\right)$$

Note that we are heavily relying on the fact that the theorem mentioned above holds for every choice of the revealed loss functions. This proves the claim. ■

Problem 4. Design an OCO algorithm attaining the same bounds as OGD, upto factors logarithmic in D and G, without knowing G and D to begin with.

Solution. In class, we have shown that OGD with step sizes $\eta_t = \frac{D}{G\sqrt{t}}$ gives the following regret bound.

$$\text{regret}_T \leq \frac{3}{2}GD\sqrt{T} = O(\sqrt{T})$$

We will now design an OCO algorithm that achieves the same asymptotic regret bound, without even knowing G and D.

So, let f be a convex function on a convex domain $\mathcal{K}$. Also, assume that there *exists* G such that $||\nabla f(\boldsymbol{x})|| \leq G$ for all $\boldsymbol{x} \in \mathcal{K}$, and *assume* that the diameter of $\mathcal{K}$ is D. Note that we are only assuming that these numbers exist, and we don't actually know their values. Also, suppose $\boldsymbol{x}_1 \in \mathcal{K}$ is the initial point.

For each $t \in [T]$, define D_t as follows.

$$D_1 = 1$$
$$D_t = \begin{cases} D_{t-1} & , \quad \text{if } ||\boldsymbol{x}_t - \boldsymbol{x}_1|| \leq D_{t-1} \\ 2D_{t-1} & , \text{ otherwise} \end{cases}$$

Similarly, for each $t \in [T]$, define G_t as follows.

$$G_1 = ||\nabla_1||$$
$$G_t = \max(G_{t-1}, ||\nabla_t||)$$

Then, we claim that with step sizes $\eta_t = \frac{D_t}{G_t\sqrt{t}}$, the usual OGD algorithm gives $\text{regret}_T \leq O(\sqrt{T})$. Let us now prove this.

As usual, let

$$\boldsymbol{x}^* = \underset{\boldsymbol{x}\in\mathcal{K}}{\operatorname{argmin}} \sum_{t=1}^{T} f_t(\boldsymbol{x})$$

First, observe that $D_1 \leq D_2 \leq \cdots \leq D_T$ and similarly $G_1 \leq G_2 \leq \cdots \leq G_T$. This is easy to see from the definitions of these sequences.

Now, we know that $\boldsymbol{x}_{t+1} = \Pi_{\mathcal{K}}(\boldsymbol{y}_{t+1})$ for each t. This implies the following.

$$||\boldsymbol{x}_{t+1} - \boldsymbol{x}^*||^2 \leq ||\boldsymbol{y}_{t+1} - \boldsymbol{x}^*||^2 = ||\boldsymbol{x}_t - \eta_t \nabla_t - \boldsymbol{x}^*||^2$$

The above is true by the Pythagorean Theorem. So, we get the following.

$$\begin{aligned} ||\boldsymbol{x}_{t+1} - \boldsymbol{x}^*||^2 &\leq ||\boldsymbol{x}_t - \eta_t \nabla_t - \boldsymbol{x}^*||^2 \\ &= ||\boldsymbol{x}_t - \boldsymbol{x}^*||^2 + \eta_t^2 ||\nabla_t||^2 - 2\eta_t \langle \nabla_t, \boldsymbol{x}_t - \boldsymbol{x}^* \rangle \end{aligned}$$

Rearranging the above inequality, we get the following.

$$2 \langle \nabla_t, \boldsymbol{x}_t - \boldsymbol{x}^* \rangle \leq \frac{||\boldsymbol{x}_t - \boldsymbol{x}^*||^2 - ||\boldsymbol{x}_{t+1} - \boldsymbol{x}^*||^2}{\eta_t} + \eta_t ||\nabla_t||^2$$

Moreover, by convexity of f_t, we know the following.

$$f_t(\boldsymbol{x}_t) - f_t(\boldsymbol{x}^*) \leq \langle \nabla_t, \boldsymbol{x}_t - \boldsymbol{x}^* \rangle$$

Combining the last two inequalities, we get the following.

$$2(f_t(\boldsymbol{x}_t) - f_t(\boldsymbol{x}^*)) \leq \frac{||\boldsymbol{x}_t - \boldsymbol{x}^*||^2 - ||\boldsymbol{x}_{t+1} - \boldsymbol{x}^*||^2}{\eta_t} + \eta_t ||\nabla_t||^2$$

Note that the above inequality is true for all $t \in [T]$. So, summing over all t, we get the following, where the convention is $1/\eta_0 = 0$ and we are using the fact that $||\boldsymbol{x}_{T+1} - \boldsymbol{x}^*|| \geq 0$.

$$\begin{aligned} 2 \cdot \text{regret}_T &\leq \sum_{t=1}^{T} \frac{||\boldsymbol{x}_t - \boldsymbol{x}^*||^2 - ||\boldsymbol{x}_{t+1} - \boldsymbol{x}^*||^2}{\eta_t} + \sum_{t=1}^{T} \eta_t ||\nabla_t||^2 \\ &\leq \sum_{t=1}^{T} ||\boldsymbol{x}_t - \boldsymbol{x}^*||^2 \left(\frac{1}{\eta_t} - \frac{1}{\eta_{t-1}} \right) + \sum_{t=1}^{T} \eta_t ||\nabla_t||^2 \\ &\leq \sum_{t=1}^{T} D^2 \left(\frac{1}{\eta_t} - \frac{1}{\eta_{t-1}} \right) + \sum_{t=1}^{T} \eta_t ||\nabla_t||^2 \\ &\leq \frac{D^2}{\eta_T} + \sum_{t=1}^{T} \frac{D_t}{G_t \sqrt{t}} G_t^2 \\ &= \frac{D^2 G_T \sqrt{T}}{D_T} + \sum_{t=1}^{T} \frac{D_t G_t}{\sqrt{t}} \\ &\leq \frac{D^2 G_T \sqrt{T}}{D_T} + D_T G_T \sum_{t=1}^{T} \frac{1}{\sqrt{t}} \\ &\leq \frac{D^2 G_T \sqrt{T}}{D_T} + 2 D_T G_T \sqrt{T} \end{aligned}$$

Above, we have used the facts that D_t, G_t are non-decreasing sequences. Now, observe that $G_T \leq G$ (because G is an upper bound on the gradients, and G_T is the maximum norm of a gradient seen till time T). So, we get that

$$2 \cdot \text{regret}_T \leq \frac{D^2 G \sqrt{T}}{D_T} + 2 D_T G \sqrt{T}$$

Now, we consider two cases.

(1) In the first case, we have $D_T \leq D$. Also, we know that $1 = D_1 \leq D_T$. So, in this case we see that
$$\frac{D^2G\sqrt{T}}{D_T} + 2D_TG\sqrt{T} \leq D^2G\sqrt{T} + 2DG\sqrt{T} = O(\sqrt{T})$$
and hence we have an $O(\sqrt{T})$ regret bound. Note that we cannot do any better than the D^2G term, because the bound must hold for all T, in particular $T = 1$. For that case, we have $D_T = 1$, and the only bound we know on $||\boldsymbol{x}_1 - \boldsymbol{x}^*||^2$ is D^2.

(2) In the second case, we have $D < D_T$. Suppose $t_0 + 1 \leq T$ is the *last time step* when the sequence D_t was updated, i.e
$$D_{t_0+1} = 2D_{t_0}$$
Clearly, we see that $D_T = D_{t_0+1} = 2D_{t_0}$. Also, by our definition, this update happened only because
$$D_{t_0} < ||\boldsymbol{x}_{t_0+1} - \boldsymbol{x}_1|| \leq D$$
So, we have that
$$D_{t_0} < D < D_T$$
which is the same as the inequality
$$\frac{D_T}{2} < D < D_T$$
In this case, we have that
$$\frac{D^2G\sqrt{T}}{D_T} + 2D_TG\sqrt{T} \leq DG\sqrt{T} + 4DG\sqrt{T} = O(\sqrt{T})$$
and hence in this case as well, we have an $O(\sqrt{T})$ regret bound.

So, in all cases the given regret bound follows, and this completes the proof of the claim. ■

Problem 5. Implement SGD for SVM training. Run the results on CIFAR-10 and also MNIST. Compare the results with offline GD algorithm. Compare the accuracies on test data.

Solution. Here is the GitHub link: https://github.com/codetalker7/ogd-vs-sgd. ■