Zeroth-Order Online Convex Optimization

Yan Pan

Carnegie Mellon University

April 27, 2023

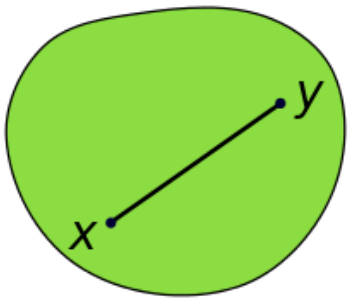
Convex Optimization

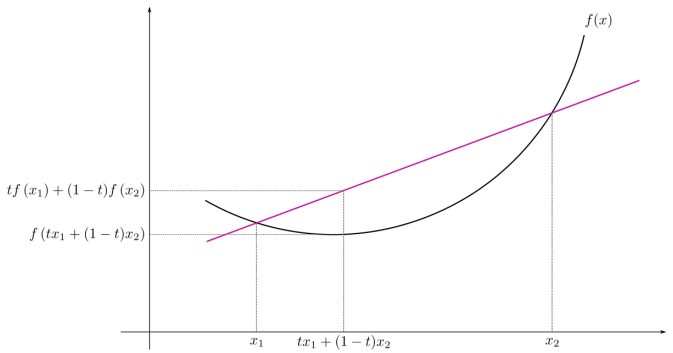
A set $\mathcal{D}$ is convex if for every $x, y \in \mathcal{D}$, $0 \leq \lambda \leq 1$, we have

$$\lambda x + (1 - \lambda) y \in \mathcal{D}.$$

A function $f : \mathcal{D} \to \mathbb{R}$ is convex if $\mathcal{D}$ is convex and for every $x, y \in \mathcal{D}$, $0 \leq \lambda \leq 1$,

$$f(\lambda x + (1 - \lambda) y) \leq \lambda f(x) + (1 - \lambda) f(y)$$





If $\nabla f(x^*) = 0$, then x^* is the only global/local minima of f.

Examples of Convex Functions

Most classical machine learning models (e.g. linear regression, logistic regression, SVM, kernel methods)

Most loss functions (e.g. mean squared loss, cross-entropy loss)

~~Neural networks (unfortunately they are not convex, but somehow many convex optimization algorithms still work; this is another interesting field of research)~~

(Stochastic) Gradient Descent

At every timestep t, update

$$x_{t+1} = x_t - \eta g_t.$$

In **gradient descent**, $g_t = \nabla f(x_t)$.

In **stochastic gradient descent**, g_t is a random vector, such that $\mathbb{E}[g_t] = \nabla f(x_t)$, and $\mathbb{E}[\|g_t\|^2] \leq G$ for some constant G.

If the constraint set $\mathcal{D}$ is a closed and bounded convex set, we can do **projected gradient descent**, where we project x_{t+1} back to $\mathcal{D}$ if it is "out of bound",

$$x_{t+1} = P_{\mathcal{D}}(x_t - \eta g_t).$$

Convergence of SGD

It is not hard to show that gradient descent converges.

We need to assume that f is convex and $\|\nabla^2 f(x)\| \leq L$ ("L-smooth").

Convergence of Gradient Descent

If f is convex, then

$$f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle.$$

Plugging into $y = x^*$, $x = x_t$, we have

$$\begin{aligned} f(x^*) &\geq f(x_t) + \langle \nabla f(x_t), x^* - x_t \rangle \\ &= f(x_t) + \frac{1}{\eta} \langle x_t - x_{t+1}, x^* - x_t \rangle. \end{aligned}$$

By "law of cosines", we get an interesting formula

$$\begin{aligned} f(x_t) &\leq f(x^*) + \frac{1}{2\eta} \left(\|x^* - x_t\|^2 - \|x^* - x_{t+1}\|^2 + \|x_t - x_{t+1}\|^2 \right) \\ &\leq f(x^*) + \frac{1}{2\eta} \left(\|x^* - x_t\|^2 - \|x^* - x_{t+1}\|^2 + \eta^2 \|g_t\|^2 \right). \end{aligned}$$

Convergence of Gradient Descent

Then, summing up from $t = 1, \ldots, T$, we have

$$\frac{1}{T}\sum_{t=1}^{T} f(x_t) \leq f(x^*) + \frac{1}{2\eta}\left(\|x^* - x_0\|^2 + \eta^2 \sum_{t=1}^{T} \|g_t\|^2\right)$$

We just need to bound $\|g_t\|^2$!

Convergence of Gradient Descent

For gradient descent, if f is L-smooth, then by Taylor's theorem,

$$f(y) \leq f(x) + \langle \nabla f(x), y - x \rangle + \frac{L}{2}\|y - x\|^2.$$

Plugging into $y = x_{t+1}$ and $x = x_t$, we have

$$\begin{aligned} f(x_{t+1}) &\leq f(x_t) + \langle \nabla f(x_t), x_{t+1} - x_t \rangle + \frac{L}{2}\|x_{t+1} - x_t\|^2 \\ &\leq f(x_t) - \eta\|\nabla f(x_t)\|^2 + \frac{L}{2}\eta^2\|\nabla f(x_t)\|^2. \end{aligned}$$

Choosing $\eta = \frac{1}{L}$, we have

$$\|\nabla f(x_t)\|^2 \leq \frac{2}{\eta}(f(x_t) - f(x_{t+1}))$$

so the sum is $O(1/\eta)$.

The average loss is at most $O(1/T)$.

Convergence Rate of Gradient Descent

For SGD, we just bound $\|g_t\|^2$ by the variance, so sum is $O(T)$.

We can pick $\eta = O(1/\sqrt{T})$ in this case, so the average loss is at most $O(1/\sqrt{T})$.

Online Convex Optimization

Instead of a single objective function, a sequence of functions $f_1, f_2, \ldots$ is given.

Need to pick a point x_t before knowing anything about f_t.

Try to minimize the regret compared to the minima x^* of the sum

$$R_T := \sum_{t=1}^{T} f_t(x_t) - \min_{x^* \in \mathcal{D}} \sum_{t=1}^{T} f_t(x^*).$$

Goal is to make $\lim_{T \to \infty} \frac{R_T}{T} \to 0$, so we have vanishing average regret.

Examples of Online Optimization

(Mini-batch) stochastic gradient descent: At every epoch, sample a batch from the dataset, and optimize the loss on the batch. This is the most widely used optimization algorithm in machine learning.

Non-convex optimization: In empirical risk minimization, to minimize $f(w) = \sum_{i=1}^{N} \ell(h(w, x_i), y_i)$, with the model h non-convex and loss ℓ convex, we can equivalently solve the following online convex optimization problem

$$f_t(w) = \frac{1}{N} \sum_{i=1}^{N} \ell(h(w_t, x_i) + \langle \nabla_w h(w_t, x_i), w - w_t \rangle, y_i).$$

Examples of Online Optimization

Multi-armed bandits: There are d machines and you can pick which one to play. After you play, you can observe the full reward vector ℓ_t. We can maintain a probability distribution p_t and define $f_t(p_t) = -\langle \ell_t, p_t \rangle$. Then, we can optimize using projected gradient descent.

Reinforcement learning: The environment changes when the player makes an action. The objective function ("value function") is dependent on the current state and the player's action.

Gradient Descent Still Works

This problem seems hard, since we don't know anything about functions in the future.

Typically, there is no guarantee that the functions are similar.

Surprisingly, gradient descent can still minimize the regret, even if we use a different function for every update step!

This is because the baseline is the minimizer of the sum $\min_{x^* \in \mathcal{D}} \sum_{t=1}^{T} f_t(x^*)$, and not the minimizers of each function.

Proof of Gradient Descent Convergence

The proof is pretty much the same as the stochastic gradient case.

We assume f_t to be Lipschitz, that is for every $x, y \in \mathcal{D}$,

$$|f(x) - f(y)| \leq L\|x - y\|$$

which bounds the gradient $\|\nabla f(x)\| \leq L$.

The regret is $O(\sqrt{T})$, so the average regret is vanishing (Zinkevich, 2003).

Zeroth-Order Online Convex Optimization

At every timestep t, instead of $\nabla f_t(x_t)$, we can only observe $f_t(x_t)$.

This is natural in scenarios where gradient is hard to obtain, or the objective function is not differentiable.

Can we still use our favorite gradient-based methods?

Revisiting an Example

A company wants to decide how much money to spend on advertising their products in d channels.

The objective function is the profit, which is changing rapidly over time, and in general there is no guarantee about how it will change.

The company definitely cannot know anything about the future, and they need to choose the allocations before observing the profit.

The gradient is hard to obtain in this case — they don't know what the objective function is. They only know their profits.

An Offline Solution

We may try to approximate $\nabla f(x)$ by

$$\langle \nabla f(x), u \rangle = \lim_{\delta \to 0} \frac{f(x + \delta u) - f(x)}{\delta}.$$

Pick a sufficiently small δ, so when f is Lipschitz, this will be a good approximation of the partial derivatives.

In $\mathbb{R}^d$, we need to approximate d times to estimate the gradient.

But in the online setting, we can only observe the value of f_t at one point! Once we observe the value, the objective function changes to f_{t+1}, and we cannot get more information about f_t.

Solution of Flaxman et al. (2005)

What if we can sample a unit random vector?

Key idea: Use Stoke's theorem to approximate the stochastic gradient.

$$\nabla f(x) \approx \mathop{\mathbb{E}}_{u \sim \mathcal{S}} \left[\frac{d}{\delta} f(x + \delta u) u \right] = \mathop{\mathbb{E}}_{u \sim \mathcal{S}} \left[\frac{d}{\delta} (f(x + \delta u) - f(x)) u \right]$$

We only need the function value at one point $f(x + \delta v)$ to approximate the stochastic gradient!

Solution of Flaxman et al. (2005)

Formally, let $\mathcal{S}$ be the unit sphere in $\mathbb{R}^d$, and $\mathcal{B}$ be the unit ball in $\mathbb{R}^d$.

By Stoke's theorem

$$\nabla \int_{\delta\mathcal{B}} f(x+v)\, dv = \int_{\delta\mathcal{S}} f(x+u)\frac{u}{\|u\|}\, du.$$

Then, since $\text{vol}(\mathcal{B}) = d\,\text{vol}(\mathcal{S})$, we have

$$\mathop{\mathbb{E}}_{u\sim\mathcal{S}}\left[\frac{d}{\delta} f(x+\delta u)u\right] = \nabla \widehat{f}(x)$$

where

$$\widehat{f}(x) = \mathop{\mathbb{E}}_{v\sim\mathcal{B}}[f(x+\delta v)].$$

Solution of Flaxman et al. (2005)

If we sample $v \sim S$, then $\frac{d}{\delta} f(x + \delta v)v$ is a stochastic gradient of $\widehat{f}$ at x, which we can use to do stochastic gradient descent.

$\widehat{f}$ is an "average" of f in a neighborhood of radius δ, so when δ is small and f is Lipschitz, the difference between f and $\widehat{f}$ is small.

Algorithm: At every timestep, sample $u_t \sim \mathcal{S}$, and update using

$$x_{t+1} = P_{\mathcal{D}}\left(x_t - \eta \frac{\delta}{d} f_t(x_t + \delta u_t)u_t\right).$$

Bounding the Regret

We can bound $\|g_t\|^2 \leq \frac{d}{\delta} \max_{x \in \mathcal{D}} |f_t(x)|$.

The standard online gradient descent proof gives us a bound on the regret of $\widehat{f}_t$ as $O(\frac{\sqrt{T}}{\delta})$.

However, this is a bound on the regret of $\widehat{f}_t$! We know that if each f_t is Lipschitz, then there is an additional error of

$$\sum_{t=1}^{T} f_t(x_t) - \sum_{t=1}^{T} \widehat{f}_t(x_t) \leq O(\delta T)$$

Choosing $\delta = O(T^{-1/4})$ gives us a regret of $O(T^{3/4})$.

Extensions of the Setting

There are various extensions of this "bandit" setting in online convex optimization.

Better algorithms exist if additional assumptions are made.

Two-Point Estimates (Agarwal and Dekel, 2010)

When we have two points as feedback, similarly we can estimate the gradient by

$$\nabla f(x) \approx \mathop{\mathbb{E}}_{u \sim \mathcal{S}} \left[\frac{d}{2\delta} (f(x + \delta u) - f(x - \delta u)) u \right].$$

When f is Lipschitz, $|f(x + \delta u) - f(x - \delta u)| \leq 2\delta$, so we removed the δ^{-1} factor in the bound of $\|g_t\|$!

Then, we can choose $\delta = O(\frac{1}{\sqrt{T}})$, so regret is bounded by $O(\sqrt{T})$, which asymptotically is as good as when we have the gradient!

Smooth Objective Functions (Saha and Tewari, 2011)

What if we sample from other than a sphere?

For a positive semidefinite matrix A, we can also approximate by

$$\nabla \widehat{f}(x) = \frac{d}{\delta} f(x + \delta A u) A^{-1} u$$

where $u \sim \mathcal{S}$ and

$$\widehat{f}(x) = \mathop{\mathbb{E}}_{u \sim \mathcal{B}}[f(x + \delta A u)].$$

Saha and Tewari (2011) improve the regret to $O(T^{2/3})$ for smooth functions by finding a sequence of A_t from a self-concordant barrier function using an interior point method.

Application in Concave Games (Bravo et al., 2018)

Suppose we have a repeated game, where each players have a concave utility function u_i.

The players can only optimize their action x_i, but they cannot control the action of other players x_{-i}.

We can define $f_t(x_{t,i}) = u_i(x_{t,i}, x_{t,-i})$, then this becomes an online optimization problem.

The individual players can use the algorithm by Flaxman et al. (2005) if they only receive their utility as feedback.

Bravo et al. (2018) shows that the game converges to a Nash equilibrium if the players use this strategy.

Agarwal, A. and Dekel, O. (2010). Optimal algorithms for online convex optimization with multi-point bandit feedback. In *COLT*, pages 28–40. Citeseer.

Bravo, M., Leslie, D., and Mertikopoulos, P. (2018). Bandit learning in concave n-person games. *Advances in Neural Information Processing Systems*, 31.

Flaxman, A. D., Kalai, A. T., and McMahan, H. B. (2005). Online convex optimization in the bandit setting: gradient descent without a gradient. In *Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 385–394.

Saha, A. and Tewari, A. (2011). Improved regret guarantees for online smooth convex optimization with bandit feedback. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 636–642. JMLR Workshop and Conference Proceedings.

Zinkevich, M. (2003). Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the 20th International Conference on Machine Learning*, pages 928–936.