First-Order Algorithms for Online Optimization and Learning

CS245: Online Optimization and Learning

Xin Liu
SIST, ShanghaiTech University

Review of Convex Optimization: Norm

Definition 1 (ℓ_p Norm)

$\|x\|_p = \left(\sum_{i=1}^n |x_i|^p\right)^{1/p}, \forall p \geq 1$ and $\|x\|_\infty = \max_{i=1,\cdots,n} |x_i|$.

Norm equivalence:

$$\|x\|_\infty \leq \|x\|_2 \leq \|x\|_1 \leq \sqrt{n}\|x\|_2 \leq n\|x\|_\infty.$$

Triangle inequality:

$$\|x + y\| \leq \|x\|_2 + \|y\|_2.$$

Cauchy-Schwarz inequality:

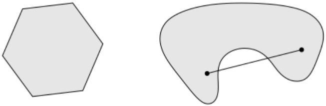
$$|\langle x, y\rangle| \leq \|x\|_2\|y\|_2.$$

Review of Convex Optimization: Convex Set

Definition 2 (Convex Set)

A set $\mathcal{K}$ is convex if $\forall x, y \in \mathcal{K}$, all the points on the line segment are also in $\mathcal{K}$, that is

$$\alpha x + (1-\alpha)y \in \mathcal{K}, \ \ \alpha \in [0,1].$$



(non)-convex sets.

Probability simplex: $\sum_{i=1}^{n} p_i = 1, \ \ p_i \geq 0, \ \ \forall i \in [n]$.

Ellipse set: $\|x\|_A = \sqrt{x^T A x} \leq 1, A \succeq 0$.

Review of Convex Optimization: Preserving convexity

Operations that preserve convexity:

- Nonnegative weighted sums:

$$g(x) = w_1 f_1(x) + w_2 f_2(x), \text{ if } w_1, w_2 \geq 0.$$

- Composition with an affine mapping:

$$g(x) = f(Ax + b).$$

- Pointwise maximum:

$$g(x) = \max\{f_1(x), f_2(x)\}.$$

- Conjugate of a function:

$$g(y) = \sup\langle y, x\rangle - f(x).$$

Review of Convex Optimization: Convex Function

Definition 3 (Convex Function)

A function $f : \mathcal{K} \to \mathbb{R}$ is convex if for any $\alpha \in [0, 1]$

$$f(\alpha x + (1-\alpha)y) \leq \alpha f(x) + (1-\alpha)f(y).$$

It is strictly convex if "$<$" holds in the inequality above.

Definition 4 (First-order condition)

If f is differentiable, that is, its gradient $\nabla f(x)$ exits $\forall x \in \mathcal{K}$, then f is convex iff

$$f(y) \geq f(x) + \langle y - x, \nabla f(x) \rangle, \forall x, y \in \mathcal{K}.$$

Definition 5 (Second-order condition)

If f is twice-differentiable, then f is convex iff

$$\nabla^2 f(x) \succeq 0, \forall x \in \mathcal{K}.$$

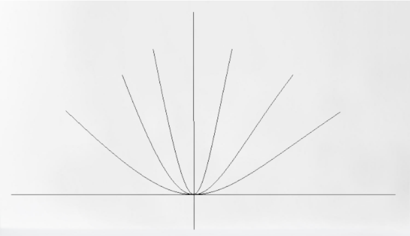
Review of Convex Optimization: Strongly Convex Function

Definition 6 (Strongly Convex Function)

A function $f : \mathcal{K} \to \mathbb{R}$ is α-strongly convex if

$$f(y) \geq f(x) + \langle y - x, \nabla f(x) \rangle + \frac{\alpha}{2}\|y - x\|^2, \ \forall x, y \in \mathcal{K}.$$

A function f is α-strongly convex iff $f(x) - \frac{\alpha}{2}\|x\|^2$ is convex. A large value of α implies a large gradient.



Strongly convex function: larger α implies large gradient.

Review of Convex Optimization: Smoothness Function

Definition 7 (Lipschitz Function)

A function $f : \mathcal{K} \to \mathbb{R}$ is Lipschitz continuous with Lipschitz constant G if

$$|f(y) - f(x)| \leq G\|y - x\|, \ \forall x, y \in \mathcal{K}.$$

Definition 8 (Smooth Function)

A function $f : \mathcal{K} \to \mathbb{R}$ is β-smoothness if

$$f(y) \leq f(x) + \langle y - x, \nabla f(x) \rangle + \frac{\beta}{2}\|y - x\|^2, \ \forall x, y \in \mathcal{K}.$$

A function f is β-smoothness is equivalent to say

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

Review of Convex Optimization: Conditional Number

Definition 9 (Conditional number of f)

A function $f : \mathcal{K} \to \mathbb{R}$ is α-strongly convex and β-smoothness. If it is twice-differentiable, its Hessian is

$$\alpha I \preceq \nabla^2 f(x) \preceq \beta I.$$

We say it is γ-well-conditioned with

$$\gamma = \frac{\alpha}{\beta} \leq 1.$$

A large γ means the function f is "better"-conditioned.

- Every "direction" is good to decrease the function (e.g., $f(x) = x^2$).
- Gradient descent algorithms will achieve a faster rate.

Review of Convex Optimization: Optimality Condition

Definition 10 (First-order Optimality of Convex function)

Given a convex and differentiable function $f : \mathcal{K} \to \mathbb{R}$, a point $x^* \in \mathcal{K}$ is optimal iff

$$\langle y - x^*, \nabla f(x^*) \rangle \geq 0, \forall y \in \mathcal{K}.$$

Any feasible direction $y - x^*$ from x^* increases the function value as follows

$$f(y) \geq f(x^*) + \langle y - x^*, \nabla f(x^*) \rangle, \ \forall y \in \mathcal{K}.$$

For a convex function, local optimal $\Longrightarrow$ global optimal.

Let $\mathcal{K} = \mathbb{R}^n$ and the optimality condition simply reduces to

$$\nabla f(x^*) = 0.$$

Convergence Rate of Gradient Descent

	general	α-strongly convex	β-smooth	γ-well conditioned
Gradient descent	$\frac{1}{\sqrt{T}}$	$\frac{1}{\alpha T}$	$\frac{\beta}{T}$	$e^{-\gamma T}$

Convergence rate of gradient descent.

An alternative measure is the iterative complexity to achieve ϵ-optimal, i.e.,

$$f(x_T) - \min_x f(x) \leq \epsilon, \forall \epsilon > 0.$$

Gradient Descent Algorithm

Gradient Descent [Cauchy 1847]

Initialization: $x_1 \in \mathcal{K}$ and step sizes $\{\eta_t\}$.
For $t = 1, \cdots, T$:

- **Gradient descent:** $y_{t+1} = x_t - \eta_t \nabla f(x_t)$.
- **Projection:** $x_{t+1} = \prod_{\mathcal{K}}(y_{t+1})$.

Intuition of GD:

$$
\begin{aligned}
x_{t+1} &= \arg\min_{x \in \mathcal{K}} \; f(x_t) + \langle x - x_t, \nabla f(x_k) \rangle + \frac{1}{2\eta_t} \|x - x_t\|^2 \\
&= \arg\min_{x \in \mathcal{K}} \; \langle x - x_t, \nabla f(x_t) \rangle + \frac{1}{2\eta_t} \|x - x_t\|^2
\end{aligned}
$$

GD is minimizing a quadratic approximation of f function at the point x_t.

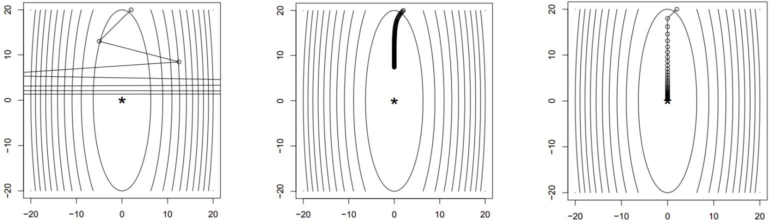
Gradient Descent

Initialization: $x_1 \in \mathcal{K}$ and step sizes $\{\eta_t\}$.

For $t = 1, \cdots, T$:

- **Gradient descent:** $y_{t+1} = x_t - \eta_t \nabla f(x_t)$.
- **Projection:** $x_{t+1} = \prod_{\mathcal{K}}(y_{t+1})$.

Learning rate is important (GD for $f(x) = 5x_1^2 + 0.5x_2^2$):



large η_t small η_t good η_t

GD for γ-well conditioned functions

Theorem 11 (Unconstrained case $\mathcal{K} = \mathbb{R}^d$)

Suppose $f : \mathbb{R}^d \to \mathbb{R}$ is a γ-well conditioned function with the minimizer x^*. Let $\eta = 1/\beta$. GD algorithm converges as

$$f(x_t) - f(x^*) \leq (f(x_1) - f(x^*))\, e^{-\gamma t}.$$

GD achieves the linear convergence:

GD for γ-well conditioned functions

Theorem 11 (Unconstrained case $\mathcal{K} = \mathbb{R}^d$)

Suppose $f : \mathbb{R}^d \to \mathbb{R}$ is a γ-well conditioned function with the minimizer x^*. Let $\eta = 1/\beta$. GD algorithm converges as

$$f(x_t) - f(x^*) \leq (f(x_1) - f(x^*))\, e^{-\gamma t}.$$

GD achieves the linear convergence:

- Learning rate is related to the "smoothness" (a smooth function can always be decreasing given a sufficient small step-size).

GD for γ-well conditioned functions

Theorem 11 (Unconstrained case $\mathcal{K} = \mathbb{R}^d$)

Suppose $f : \mathbb{R}^d \to \mathbb{R}$ is a γ-well conditioned function with the minimizer x^*. Let $\eta = 1/\beta$. GD algorithm converges as

$$f(x_t) - f(x^*) \leq (f(x_1) - f(x^*))\, e^{-\gamma t}.$$

GD achieves the linear convergence:

- Learning rate is related to the "smoothness" (a smooth function can always be decreasing given a sufficient small step-size).
- Iteration complexity is exponentially small $\log(1/\epsilon)$!

GD for γ-well conditioned functions

Theorem 11 (Unconstrained case $\mathcal{K} = \mathbb{R}^d$)

Suppose $f : \mathbb{R}^d \to \mathbb{R}$ is a γ-well conditioned function with the minimizer x^*. Let $\eta = 1/\beta$. GD algorithm converges as

$$f(x_t) - f(x^*) \leq (f(x_1) - f(x^*))\, e^{-\gamma t}.$$

GD achieves the linear convergence:

- Learning rate is related to the "smoothness" (a smooth function can always be decreasing given a sufficient small step-size).
- Iteration complexity is exponentially small $\log(1/\epsilon)$!
- GD is dimensional-free!

GD for γ-well conditioned functions – proof

A "potential/Lyapunov drift" style of analysis: define

$$\phi_t = f(x_t) - f(x^*),$$

and study the drfit

$$\phi_{t+1} - \phi_t.$$

GD for γ-well conditioned functions – proof

GD for β-smoothness functions - a reduction method

Gradient Descent for β-smoothness function

Initialization: $x_1 \in \mathcal{K}$, $\{\eta_t\}$ and $\tilde{f}(x) = f(x) + \delta\|x\|^2$.
For $t = 1, \cdots, T$:

- **Gradient descent:** $x_{t+1} = x_t - \eta_t \nabla \tilde{f}(x_t)$.

Theorem 12

Assume $\|x - y\| \le D, \forall x, y \in \mathcal{K}$. Suppose $f : \mathbb{R}^d \to \mathbb{R}$ is a smooth convex function. Let $\eta_t = \frac{1}{\beta}$ and $\delta = \frac{\beta \log t}{R^2 t}$. GD algorithm converges as

$$f(x_{t+1}) - f(x^*) = O\left(\frac{\beta \log t}{t}\right).$$

GD for β-smoothness functions – proof

GD for α-strongly convex functions - a reduction method

Gradient Descent for α-strongly convex functions

Initialization: x_1, $\{\eta_t\}$, and $\tilde{f}(x) = \mathbb{E}_{v \in \text{Unif Ball}}[f(x + \delta \mathbf{v})]$.
For $t = 1, \cdots, T$:

- **Gradient descent:** $x_{t+1} = x_t - \eta_t \nabla \tilde{f}(x_t)$.

Theorem 13

Assume $\|x - y\| \le D, \forall x, y \in \mathcal{K}$. Suppose $f : \mathbb{R}^d \to \mathbb{R}$ is α strongly convex function. Let $\delta = O(\frac{\log t}{t})$. GD algorithm converges as

$$f(x_{t+1}) - f(x^*) = O\left(\frac{\log t}{\alpha t}\right).$$

GD for general convex functions

Gradient Descent Algorithm

Initialization: $x_1 \in \mathcal{K}$. Choose step sizes $\{\eta_t\}$ satisfying $\sum_{t=1}^{\infty} \eta_t^2 < \infty$ and $\sum_{t=1}^{\infty} \eta_t = \infty$.

For $t = 1, \cdots, T$:

- **Gradient descent:** $y_{t+1} = \prod_{\mathcal{K}}(x_t - \eta_t \nabla f(x_t))$.

Diminishing step sizes (square summable but not summable): the step sizes go to zero, but not too fast.

Theorem 14

Assume $\|x - y\| \leq D, \forall x, y \in \mathcal{K}$ and $\|\nabla f(x)\| \leq G, \forall x \in \mathcal{K}$. Suppose $f : \mathbb{R}^d \to \mathbb{R}$ is a convex function with the minimizer x^*. GD algorithm converges as

$$\min_{t \in [T]} f(x_t) - f(x^*) \leq \frac{D^2 + G^2 \sum_{t=1}^{T} \eta_t^2}{2 \sum_{t=1}^{T} \eta_t}.$$

GD for general convex functions – proof

A "potential/Lyapunov drift" style of analysis: define

$$\phi_t = \|x_t - x^*\|^2,$$

and study the drift

$$\phi_{t+1} - \phi_t.$$

Learning as Optimization – Linear Regression

Consider linear regression (LR) for "regression" (e.g., Shanghai Putong house price prediction).

Given historical/batch data $\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \cdots (\mathbf{x}_N, y_N)\}$, we do LR

$$\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{X}^T\mathbf{w} - \mathbf{y}\|_2^2 + \frac{\lambda}{2}\|\mathbf{w}\|^2$$

Gradient Descent for LR

Initialization: $w_1 \in \mathcal{K}$ and step sizes $\{\eta_t\}$.

For $t = 1, \cdots, T$:

- **Compute gradient:** $\nabla f(w_t) = \mathbf{X}\mathbf{X}^T\mathbf{w}_t - \mathbf{X}\mathbf{y} + \lambda\mathbf{w}_t$
- **Update:** $\mathbf{w}_{t+1} = \mathbf{w}_t - \eta_t \nabla f(\mathbf{w}_t)$.

Output w_T.

Learning as Optimization – Supported Vector Machine

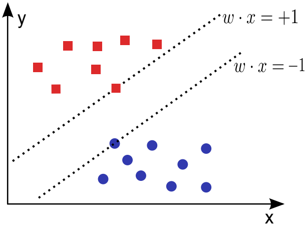
Consider Supported Vector Machine (SVM) for "classification" (e.g., spam email detection).

Given historical/batch data $\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \cdots (\mathbf{x}_N, y_N)\}$, we need to minimize the # of mistakes

$$\min_{\mathbf{w}} \sum_{n=1}^{N} \mathbb{I}(sign(\langle \mathbf{w}, \mathbf{x}_n \rangle) \neq y_n).$$

We need to do a bit relaxation:

$$\min_{\mathbf{w}} \|w\|^2$$
$$\text{s.t. } y_n \cdot \langle \mathbf{w}, \mathbf{x}_n \rangle \geq 1, \forall n \in [N].$$



We need to do a bit relaxation:

$$\min_{\mathbf{w}} \|w\|^2$$
$$\text{s.t. } y_n \cdot \langle \mathbf{w}, \mathbf{x}_n \rangle \geq 1, \forall n \in [N].$$

We want an unconstrained problem:

$$\min_{\mathbf{w}} \frac{\lambda}{N} \sum_{n=1}^{N} \max(0, 1 - y_n \cdot \langle \mathbf{w}, \mathbf{x}_n \rangle) + \frac{1}{2}\|w\|^2$$

SubGradient Descent for SVM

Initialization: $w_1 \in \mathcal{K}$ and step sizes $\eta_t = O(1/t)$.

For $t = 1, \cdots, T$:

- **Compute gradient:** $\nabla f(w_t) = -\frac{\lambda}{N} \sum_{n=1}^{N} y_n \cdot \mathbf{x}_n + \mathbf{w}_t$ if $y_n \cdot \langle \mathbf{x}_n, \mathbf{w} \rangle < 1$; otherwise $\nabla f(w_t) = w_t$.
- **Update:** $w_{t+1} = w_t - \eta_t \nabla f(w_t)$.

Output w_T or a weighted version of $\{w_t\}$.

From Offline to Online Convex Optimization

From offline to online convex optimization:

- In offline convex optimization, $f(\cdot)$ is known in advance and fixed all the time!
- In online convex optimization, $f_t(\cdot)$ is revealed after our action x_t. $\{f_t\}$ could be arbitrary, for example, it could be fixed, i.i.d., or even adversarial!

From Offline to Online Convex Optimization

From offline to online convex optimization:

- In offline convex optimization, $f(\cdot)$ is known in advance and fixed all the time!
- In online convex optimization, $f_t(\cdot)$ is revealed after our action x_t. $\{f_t\}$ could be arbitrary, for example, it could be fixed, i.i.d., or even adversarial!

From the convergence rate to regret:

From Offline to Online Convex Optimization

From offline to online convex optimization:

- In offline convex optimization, $f(\cdot)$ is known in advance and fixed all the time!
- In online convex optimization, $f_t(\cdot)$ is revealed after our action x_t. $\{f_t\}$ could be arbitrary, for example, it could be fixed, i.i.d., or even adversarial!

From the convergence rate to regret:

- For a general function $f(\cdot)$ in offline convex optimization, GD achieves $f(x_T) - f(x^*) = O(1/\sqrt{T})$.

From Offline to Online Convex Optimization

From offline to online convex optimization:

- In offline convex optimization, $f(\cdot)$ is known in advance and fixed all the time!
- In online convex optimization, $f_t(\cdot)$ is revealed after our action x_t. $\{f_t\}$ could be arbitrary, for example, it could be fixed, i.i.d., or even adversarial!

From the convergence rate to regret:

- For a general function $f(\cdot)$ in offline convex optimization, GD achieves $f(x_T) - f(x^*) = O(1/\sqrt{T})$.
- For a sequence of general function $\{f_t\}$ in online convex optimization, online GD achieves

$$\sum_{t=1}^{T} f_t(x_t) - f_t(x^*) = O(\sqrt{T}) \ ?$$

Online Gradient Descent (OGD)

Initialization: $x_1 \in \mathcal{K}$ and $\{\eta_t\}$.
For $t = 1, \cdots, T$:

- **Learner:** Submit x_t.
- **Environment:** Observe the convex loss $f_t(\cdot)$.
- **Update:** $x_{t+1} = \prod_{\mathcal{K}}(x_t - \eta_t \nabla f_t(x_t))$.

The intuition of OGD is to approximate/predict $f_{t+1}(x)$ with $\hat{f}_{t+1}(x)$ as following:

$$\hat{f}_{t+1}(x) = f_t(x_t) + \langle x - x_t, \nabla f_t(x_t) \rangle + \frac{1}{2\eta_t} \|x - x_t\|^2.$$

Online Gradient Descent

The regret of OGD is:

$$\mathrm{Regret}(T) = \sum_{t=1}^{T} f_t(x_t) - \min_{x \in \mathcal{K}} \sum_{t=1}^{T} f_t(x).$$

Theorem 15

Assume $\|x - y\| \leq D, \forall x, y \in \mathcal{K}$ and $\|\nabla f_t(x)\| \leq G, \forall x \in \mathcal{K}$ for any t. Let $\eta_t = \frac{D}{G\sqrt{t}}$. OGD algorithm achieves

$$\mathit{Regret}(T) \leq \frac{3}{2} GD\sqrt{T}.$$

OGD achieves $O(\sqrt{T})$ regret:

Online Gradient Descent

The regret of OGD is:

$$\text{Regret}(T) = \sum_{t=1}^{T} f_t(x_t) - \min_{x \in \mathcal{K}} \sum_{t=1}^{T} f_t(x).$$

Theorem 15

Assume $\|x - y\| \leq D, \forall x, y \in \mathcal{K}$ *and* $\|\nabla f_t(x)\| \leq G, \forall x \in \mathcal{K}$ *for any* t. *Let* $\eta_t = \frac{D}{G\sqrt{t}}$. *OGD algorithm achieves*

$$\text{Regret}(T) \leq \frac{3}{2} GD\sqrt{T}.$$

OGD achieves $O(\sqrt{T})$ regret:

- Learning rate is time-varying and independent with time horizon T (note learning rate is extremely important).

Online Gradient Descent

The regret of OGD is:

$$\text{Regret}(T) = \sum_{t=1}^{T} f_t(x_t) - \min_{x\in\mathcal{K}} \sum_{t=1}^{T} f_t(x).$$

Theorem 15

Assume $\|x-y\| \le D, \forall x, y \in \mathcal{K}$ and $\|\nabla f_t(x)\| \le G, \forall x \in \mathcal{K}$ for any t. Let $\eta_t = \frac{D}{G\sqrt{t}}$. OGD algorithm achieves

$$\mathit{Regret}(T) \le \frac{3}{2} GD\sqrt{T}.$$

OGD achieves $O(\sqrt{T})$ regret:

- Learning rate is time-varying and independent with time horizon T (note learning rate is extremely important).
- GD is dimensional-free but it is related to D and G.

Online Gradient Descent – Proof

Similar with the gradient descent for general convex functions, we use a "potential/Lyapunov drift" style of analysis: define

$$\phi_t = \|x_t - x^*\|^2,$$

and study the drift

$$\phi_{t+1} - \phi_t.$$

Lower Bounds for Online Convex Optimization

Along with the "style" of this course, we justify if $O(DG\sqrt{T})$ achieved by online gradient descent can be improvable?

- Theorem 15 does not assume any good properties on the loss functions $\{f_t\}$.
- Scaling with D and G is quite standard. How about $\sqrt{T}$?

Lower Bounds for Online Convex Optimization

Along with the "style" of this course, we justify if $O(DG\sqrt{T})$ achieved by online gradient descent can be improvable?

- Theorem 15 does not assume any good properties on the loss functions $\{f_t\}$.
- Scaling with D and G is quite standard. How about $\sqrt{T}$?

We need to investigate what is the lower bound for a general online convex optimization problem:

- Given an OCO problem $\mathcal{P}$, any online algorithms will incur at least $\Omega(\sqrt{T})$ regret?
- We design OCO problems instead of algorithms.

Along with the "style" of this course, we justify if $O(DG\sqrt{T})$ achieved by online gradient descent can be improvable?

- Theorem 15 does not assume any good properties on the loss functions $\{f_t\}$.
- Scaling with D and G is quite standard. How about $\sqrt{T}$?

We need to investigate what is the lower bound for a general online convex optimization problem:

- Given an OCO problem $\mathcal{P}$, any online algorithms will incur at least $\Omega(\sqrt{T})$ regret?
- We design OCO problems instead of algorithms.

OCO problems $\Longrightarrow$ The best algorithms $\Longrightarrow$ Min upper bounds.
Online algorithms $\Longrightarrow$ The hardest OCO problems $\Longrightarrow$ Max lower bounds.

Lower Bounds for Online Convex Optimization

Design an OCO problem $\mathcal{P}$ means to design an sequence of $\{f_t\}$ s.t.

$$\max_{\{f_t\}} \text{Regret}(T)$$

is maximized for any online algorithms. It seems very challenging, right?

Lower Bounds for Online Convex Optimization

Design an OCO problem $\mathcal{P}$ means to design an sequence of $\{f_t\}$ s.t.

$$\max_{\{f_t\}} \text{Regret}(T)$$

is maximized for any online algorithms. It seems very challenging, right?

Let's consider a related easy problem where $\{f_t\}$ is i.i.d., we have

$$\max_{\{f_t\}} \text{Regret}(T) \geq \mathop{\mathbb{E}}_{\{f_t\}} [\text{Regret}(T)].$$

Lower Bounds for Online Convex Optimization

Design an OCO problem $\mathcal{P}$ means to design an sequence of $\{f_t\}$ s.t.

$$\max_{\{f_t\}} \mathrm{Regret}(T)$$

is maximized for any online algorithms. It seems very challenging, right?

Let's consider a related easy problem where $\{f_t\}$ is i.i.d., we have

$$\max_{\{f_t\}} \mathrm{Regret}(T) \geq \mathop{\mathbb{E}}_{\{f_t\}}[\mathrm{Regret}(T)].$$

Construct the lower bound by the probabilistic method:

$$\mathop{\mathbb{E}}_{\{f_t\}}[\mathrm{Regret}(T)] = \mathop{\mathbb{E}}_{\{f_t\}}\left[\sum_{t=1}^{T} f_t(x_t) - \min_{x\in\mathcal{K}} \sum_{t=1}^{T} f_t(x)\right].$$

Lower Bounds for Online Convex Optimization

Theorem 16

There exists an sequence of $\{f_t\}$ such that for any online algorithms it incurs at least $\Omega(\sqrt{T})$ regret.

We consider an i.i.d. sequence of linear functions $\{f_t\}$

$$f_t(x) = \langle v_t, x \rangle, \quad \|x\|_1 = 1,$$

where each element in v_t is Rademacher random variable, and we study

$$\begin{aligned}
\mathop{\mathbb{E}}_{\{f_t\}}[\text{Regret}(T)] &= \mathop{\mathbb{E}}_{\{f_t\}}\left[\sum_{t=1}^{T} f_t(x_t) - \min_{x \in \mathcal{K}} \sum_{t=1}^{T} f_t(x)\right] \\
&= \mathop{\mathbb{E}}_{\{v_t\}}\left[\sum_{t=1}^{T} \langle v_t, x_t \rangle - \min_{x \in \mathcal{K}} \sum_{t=1}^{T} \langle v_t, x \rangle\right]
\end{aligned}$$

Lower Bounds for Online Convex Optimization – Proof

From Online to Offline Convex Optimization

From online to offline convex optimization:

- In online convex optimization, choose x_t given the history until t.
- In offline convex optimization, choose x_t given f.

From Online to Offline Convex Optimization

From online to offline convex optimization:

- In online convex optimization, choose x_t given the history until t.
- In offline convex optimization, choose x_t given f.

From regret to the convergence rate:

- Online GD achieves

$$\sum_{t=1}^{T} f_t(x_t) - \min_{x \in \mathcal{K}} \sum_{t=1}^{T} f_t(x) = O(\sqrt{T})$$

with an sequence of $\{x_t\}$.

From Online to Offline Convex Optimization

From online to offline convex optimization:

- In online convex optimization, choose x_t given the history until t.
- In offline convex optimization, choose x_t given f.

From regret to the convergence rate:

- Online GD achieves

$$\sum_{t=1}^{T} f_t(x_t) - \min_{x \in \mathcal{K}} \sum_{t=1}^{T} f_t(x) = O(\sqrt{T})$$

  with an sequence of $\{x_t\}$.

- Can we use $\{x_t\}$ to produce an action $\bar{x}_T$ such that

$$f(\bar{x}_T) - f(x^*) = O(1/\sqrt{T}).$$

From Online to Offline Convex Optimization

Online Gradient Descent for a known function g

Initialization: x_1 and $\{\eta_t\}$.

For $t = 1, \cdots, T$:

- **Learner:** Submit x_t.
- **Environment:** Observe the convex loss $f_t(x_t) = g(x_t)$.
- **Update:** $x_{t+1} = x_t - \eta_t \nabla f_t(x_t)$.

Output: $\bar{x}_T = \frac{1}{T} \sum_{t=1}^{T} x_t$

Theorem 17

Given an sequence of $\{x_t\}$ returned by online gradient descent and x^* is the optimal solution to g, we have

$$g(\bar{x}_T) - g(x^*) = O(1/\sqrt{T}).$$

From Online to Offline Convex Optimization – Proof

From Online to Stochastic Convex Optimization

Online Gradient Descent for an estimated function g

Initialization: x_1 and $\{\eta_t\}$.
For $t = 1, \cdots, T$:

- **Learner:** Submit x_t.
- **Environment:** Observe the estimated $\tilde{\nabla} g(x_t)$ and the "virtual" loss $f_t(x) = \langle \tilde{\nabla} g(x_t), x \rangle$.
- **Update:** $x_{t+1} = x_t - \eta_t \nabla f_t(x_t)$.

Output: $\bar{x}_T = \frac{1}{T} \sum_{t=1}^{T} x_t$

Theorem 18

Given an sequence of $\{x_t\}$ returned by online gradient descent and x^* is the optimal solution to g, we have

$$\mathbb{E}[g(\bar{x}_T)] - g(x^*) = O(1/\sqrt{T}).$$

From Online to Stochastic Convex Optimization – Proof

Learning as Stochastic Optimization – Linear Regression

Consider linear regression (LR) for "regression" (e.g., Shanghai Putong house price prediction).

Given historical/batch data $\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \cdots (\mathbf{x}_N, y_N)\}$, we do LR

$$\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{X}^T\mathbf{w} - \mathbf{y}\|_2^2 + \frac{\lambda}{2}\|\mathbf{w}\|^2$$

Stochastic Gradient Descent for LR

Initialization: w_1 and step sizes $\{\eta_t\}$.
For $t = 1, \cdots, T$:

- **Random pick an sample:** $(\mathbf{x}_i, y_i)$
- **Compute gradient:** $\tilde{\nabla} f_t(w_t) = \mathbf{x}_i\mathbf{x}_i^T\mathbf{w}_t - \mathbf{x}_i\mathbf{y}_i + \lambda\mathbf{w}_t$
- **Update:** $\mathbf{w}_{t+1} = \mathbf{w}_t - \eta_t\tilde{\nabla} f_t(\mathbf{w}_t)$.

Output $\bar{w}_T = \frac{1}{T}\sum_{t=1}^{T} w_t$.

Online Gradient Descent - Beyond $O(\sqrt{T})$

The regret of OGD is $\text{Regret}(T) = O(\sqrt{T})$, not improvable given the lower bound of $\Omega(\sqrt{T})$. In fact, we can achieve a smaller regret for strongly convex functions.

Theorem 19

Assume $\|x - y\| \leq D, \forall x, y \in \mathcal{K}$ and α-strongly convex functions $\{f_t\}$ with $\|\nabla f_t(x)\| \leq G, \forall x \in \mathcal{K}$ for any t. Let $\eta_t = \frac{1}{\alpha t}$. OGD algorithm achieves

$$\textit{Regret}(T) \leq \frac{G^2}{2\alpha}(1 + \log(T)).$$

OGD achieves $O(\log T)$ regret:

- Learning rate is time-varying and becomes $O(1/t)$ instead of $O(1/\sqrt{t})$.

Online Gradient Descent - Beyond $O(\sqrt{T})$ – Proof

We use a "potential/Lyapunov drift" style of analysis: define

$$\phi_t = \|x_t - x^*\|^2,$$

and study the drift

$$\begin{aligned}
\phi_{t+1} - \phi_t =& \|x_{t+1} - x^*\|^2 - \|x_t - x^*\|^2 \\
=& \|x_t - \eta_t \nabla f_t(x_t) - x^*\|^2 - \|x_t - x^*\|^2 \\
=& 2\eta_t \langle x^* - x_t, \nabla f_t(x_t)\rangle + \eta_t^2 \|\nabla f_t(x_t)\|^2
\end{aligned}$$

which implies

$$\langle x_t - x^*, \nabla f_t(x_t)\rangle \le \frac{1}{2\eta_t}\left(\|x_t - x^*\|^2 - \|x_{t+1} - x^*\|^2\right) + \frac{\eta_t}{2}\|\nabla f_t(x_t)\|^2$$

If f_t is a α-strongly convex function, we have

$$f_t(x_t) - f_t(x^*) + \frac{\alpha}{2}\|x_t - x^*\|^2 \le \langle x_t - x^*, \nabla f_t(x_t)\rangle.$$

Telescope sum from $t = 1, 2, \cdots T$, we have

$$\mathrm{Regret}(T) + \sum_{t=1}^{T} \frac{\alpha}{2}\|x_t - x^*\|^2$$

$$\le \sum_{t=1}^{T} \frac{1}{2\eta_t}\left(\|x_t - x^*\|^2 - \|x_{t+1} - x^*\|^2\right) + \sum_{t=1}^{T} \frac{\eta_t}{2}\|\nabla f_t(x_t)\|^2,$$

which implies

$$2\mathrm{Regret}(T) \le \sum_{t=1}^{T} \left(\frac{1}{\eta_t} - \frac{1}{\eta_{t-1}} - \alpha\right)\|x_t - x^*\|^2 + \sum_{t=1}^{T} \eta_t\|\nabla f_t(x_t)\|^2.$$

Let $\eta_t = \frac{1}{\alpha t}$. Finally, we have

$$\text{Regret}(T) \le \sum_{t=1}^{T} \frac{\|\nabla f_t(x_t)\|^2}{2\alpha t} \le \frac{G^2}{2\alpha}(1 + \log(T)). \qquad \square$$

Online GD with carefully choosing learning rates $\{\eta_t\}$ achieves the regret:

- $O(\sqrt{T})$ if $\{f_t\}$ is convex.
- $O(\log T)$ if $\{f_t\}$ is α-strongly convex.

Online Gradient Descent - Beyond $O(\sqrt{T})$ – Proof

Let $\eta_t = \frac{1}{\alpha t}$. Finally, we have

$$\text{Regret}(T) \le \sum_{t=1}^{T} \frac{\|\nabla f_t(x_t)\|^2}{2\alpha t} \le \frac{G^2}{2\alpha}(1 + \log(T)). \qquad \square$$

Online GD with carefully choosing learning rates $\{\eta_t\}$ achieves the regret:

- $O(\sqrt{T})$ if $\{f_t\}$ is convex.
- $O(\log T)$ if $\{f_t\}$ is α-strongly convex.

How about some of functions in $\{f_t\}$ are convex and others are α-strongly convex?

- Can we achieve somethings between $O(\log T)$ and $O(\sqrt{T})$?

Adaptive Online Gradient Descent

Adaptive Online GD for Partial Strongly Convex $\{f_t\}$

Initialization: x_1.

For $t = 1, \cdots, T$:

- **Learner:** Submit x_t.
- **Environment:** Observe $f_t(x)$ with α_t-strongly convexity.
- **Update:** $\eta_t = ???, \quad x_{t+1} = x_t - \eta_t \nabla f_t(x_t)$.

Adaptive Online Gradient Descent

Adaptive Online GD for Partial Strongly Convex $\{f_t\}$

Initialization: x_1.

For $t = 1, \cdots, T$:

- **Learner:** Submit x_t.
- **Environment:** Observe $f_t(x)$ with α_t-strongly convexity.
- **Update:** $\eta_t = 1/\sum_{s=1}^{t} \alpha_s$, $\quad x_{t+1} = x_t - \eta_t \nabla f_t(x_t)$.

Adaptive Online Gradient Descent

Adaptive Online GD for Partial Strongly Convex $\{f_t\}$

Initialization: x_1.

For $t = 1, \cdots, T$:

- **Learner:** Submit x_t.
- **Environment:** Observe $f_t(x)$ with α_t-strongly convexity.
- **Update:** $\eta_t = 1/\sum_{s=1}^{t} \alpha_s$, $\quad x_{t+1} = x_t - \eta_t \nabla f_t(x_t)$.

Theorem 20

Assume $\|x - y\| \leq D, \forall x, y \in \mathcal{K}$ and convex functions $\{f_t\}$ with $\|\nabla f_t(x)\| \leq G_t, \forall x \in \mathcal{K}$ for any t. OGD algorithm above achieves

$$Regret(T) \leq \sum_{t=1}^{T} \frac{G_t^2}{2\sum_{s=1}^{t} \alpha_s}.$$

Adaptive Online Gradient Descent

Is it a good adaptive bound?

$$\text{Regret}(T) \leq \sum_{t=1}^{T} \frac{G_t^2}{2\sum_{s=1}^{t} \alpha_s}.$$

Discussion:

Adaptive Online Gradient Descent

Is it a good adaptive bound?

$$\text{Regret}(T) \leq \sum_{t=1}^{T} \frac{G_t^2}{2\sum_{s=1}^{t} \alpha_s}.$$

Discussion:

- $O(\log T)$ if $\{f_t\}$ are α-strongly convex.

Adaptive Online Gradient Descent

Is it a good adaptive bound?

$$\text{Regret}(T) \leq \sum_{t=1}^{T} \frac{G_t^2}{2\sum_{s=1}^{t} \alpha_s}.$$

Discussion:

- $O(\log T)$ if $\{f_t\}$ are α-strongly convex.
- How about the first half of $\{f_t\}$ are strongly convex and the second half of $\{f_t\}$ are only convex?

Adaptive Online Gradient Descent

Is it a good adaptive bound?

$$\text{Regret}(T) \leq \sum_{t=1}^{T} \frac{G_t^2}{2\sum_{s=1}^{t} \alpha_s}.$$

Discussion:

- $O(\log T)$ if $\{f_t\}$ are α-strongly convex.
- How about the first half of $\{f_t\}$ are strongly convex and the second half of $\{f_t\}$ are only convex?
- How about the first half of $\{f_t\}$ are only convex and the second half of $\{f_t\}$ are strongly convex?

Adaptive Online Gradient Descent

Add regularizars to make it strongly-convex!!!

$$\tilde{f}_t(x) = f_t(x) + \frac{\lambda_t}{2}\|x\|^2.$$

Adaptive Online Gradient Descent

Add regularizars to make it strongly-convex!!!

$$\tilde{f}_t(x) = f_t(x) + \frac{\lambda_t}{2}\|x\|^2.$$

From Theorem 20, now we have the regret for $\{\tilde{f}_t\}$ functions

$$\widetilde{\text{Regret}}(T) \leq \sum_{t=1}^{T} \frac{G_t^2}{2\sum_{s=1}^{t}(\lambda_s + \alpha_s)},$$

Adaptive Online Gradient Descent

Add regularizars to make it strongly-convex!!!

$$\tilde{f}_t(x) = f_t(x) + \frac{\lambda_t}{2}\|x\|^2.$$

From Theorem 20, now we have the regret for $\{\tilde{f}_t\}$ functions

$$\widetilde{\text{Regret}}(T) \le \sum_{t=1}^{T} \frac{G_t^2}{2\sum_{s=1}^{t}(\lambda_s + \alpha_s)},$$

which implies (assuming $D = 1$)

$$2\text{Regret}(T) \le \sum_{t=1}^{T} \lambda_t + \sum_{t=1}^{T} \frac{G_t^2}{\sum_{s=1}^{t}(\lambda_s + \alpha_s)}.$$

Adaptive Online Gradient Descent

Let's look at

$$H_T(\lambda_1, \cdots, \lambda_T) := \sum_{t=1}^{T} \lambda_t + \sum_{t=1}^{T} \frac{G_t^2}{\sum_{s=1}^{t}(\lambda_s + \alpha_s)}.$$

Adaptive Online Gradient Descent

Let's look at

$$H_T(\lambda_1, \cdots, \lambda_T) := \sum_{t=1}^{T} \lambda_t + \sum_{t=1}^{T} \frac{G_t^2}{\sum_{s=1}^{t}(\lambda_s + \alpha_s)}.$$

A surprising result from [Bartlett, Hazan, and Rakhlin] [1] is if

$$\lambda_t = G_t^2 / \sum_{s=1}^{t}(\lambda_s + \alpha_s),$$

then

$$H_T(\lambda_1, \cdots, \lambda_T) \le 2 \min_{\lambda_i \ge 0} H_T(\lambda_1, \cdots, \lambda_T).$$

[1] Peter L. Bartlett, Elad Hazan, and Alexander Rakhlin. Adaptive online gradient descent. In Neural Information Processing Systems (NIPS), 2007.

Adaptive Online Gradient Descent

We enventully have

$$
\begin{aligned}
\text{Regret}(T) &\leq \min_{\lambda_i \geq 0} H_T(\lambda_1, \cdots, \lambda_T) \\
&\leq \min_{\lambda_i \geq 0} \left[\sum_{t=1}^{T} \lambda_t + \sum_{t=1}^{T} \frac{G_t^2}{\sum_{s=1}^{t} (\lambda_s + \alpha_s)} \right]
\end{aligned}
$$

Discussion:

Adaptive Online Gradient Descent

We enventully have

$$\begin{aligned}\text{Regret}(T) &\le \min_{\lambda_i \ge 0} H_T(\lambda_1, \cdots, \lambda_T) \\ &\le \min_{\lambda_i \ge 0} \left[\sum_{t=1}^{T} \lambda_t + \sum_{t=1}^{T} \frac{G_t^2}{\sum_{s=1}^{t} (\lambda_s + \alpha_s)} \right]\end{aligned}$$

Discussion:

- $O(\sqrt{T})$ is achieved with $\lambda_1 = \sqrt{T}$ and $\lambda_t = 0, \forall t \ge 2$.

Adaptive Online Gradient Descent

We enventully have

$$\begin{aligned} \text{Regret}(T) &\leq \min_{\lambda_i \geq 0} H_T(\lambda_1, \cdots, \lambda_T) \\ &\leq \min_{\lambda_i \geq 0} \left[\sum_{t=1}^{T} \lambda_t + \sum_{t=1}^{T} \frac{G_t^2}{\sum_{s=1}^{t} (\lambda_s + \alpha_s)} \right] \end{aligned}$$

Discussion:

- $O(\sqrt{T})$ is achieved with $\lambda_1 = \sqrt{T}$ and $\lambda_t = 0, \forall t \geq 2$.
- $O(\log T)$ is achieved with $\lambda_t = 0$ if $\alpha_t > 0, \forall t \geq 1$.

Adaptive Online Gradient Descent

Recall in general convex functions, we have online gradient descent with the learning rate such that

$$
\begin{aligned}
2\text{Regret}(T) &\leq \sum_{t=1}^{T} \left(\frac{1}{\eta_t} - \frac{1}{\eta_{t-1}} \right) \|x_t - x^*\|^2 + \sum_{t=1}^{T} \eta_t \|\nabla f_t(x_t)\|^2 \\
&\leq \frac{1}{\eta_T} + \sum_{t=1}^{T} \eta_t \|\nabla f_t(x_t)\|^2
\end{aligned}
$$

Assuming a fixed learning rate $\eta_t = \eta, \forall t$, we minimize the upper bound by setting

$$
\eta = \frac{1}{\sqrt{\sum_{t=1}^{T} \|\nabla f_t(x_t)\|^2}}.
$$

Adaptive Online Gradient Descent

The regret becomes "adaptive" to gradients of functions:

$$2\text{Regret}(T) \leq 2\sqrt{\sum_{t=1}^{T} \|\nabla f_t(x_t)\|^2}$$

However, the learning rate η requires all the future gradients.
Can we try the learning rate without any future information?

Adaptive Online Gradient Descent

The regret becomes "adaptive" to gradients of functions:

$$2\text{Regret}(T) \leq 2\sqrt{\sum_{t=1}^{T} \|\nabla f_t(x_t)\|^2}$$

However, the learning rate η requires all the future gradients.
Can we try the learning rate without any future information?

$$\eta_t = \frac{1}{\sqrt{\sum_{s=1}^{t} \|\nabla f_s(x_s)\|^2}} \ ?$$

The regret becomes "adaptive" to gradients of functions:

$$2\text{Regret}(T) \leq 2\sqrt{\sum_{t=1}^{T} \|\nabla f_t(x_t)\|^2}$$

However, the learning rate η requires all the future gradients.
Can we try the learning rate without any future information?

$$\eta_t = \frac{1}{\sqrt{\sum_{s=1}^{t} \|\nabla f_s(x_s)\|^2}} \; ?$$

Now the regret becomes

$$2\text{Regret}(T) \leq \frac{1}{\eta_T} + \sum_{t=1}^{T} \frac{\|\nabla f_t(x_t)\|^2}{\sqrt{\sum_{s=1}^{t} \|\nabla f_s(x_s)\|^2}}.$$

Adaptive Online Gradient Descent

A bit surprising result (verify it by yourself):

$$\sum_{t=1}^{T} \frac{\|\nabla f_t(x_t)\|^2}{\sqrt{\sum_{s=1}^{t} \|\nabla f_s(x_s)\|^2}} \leq 2\sqrt{\sum_{t=1}^{T} \|\nabla f_t(x_t)\|^2}.$$

Finally, we achieve an adaptive regret without any future information:

$$\text{Regret}(T) \leq \frac{3}{2}\sqrt{\sum_{t=1}^{T} \|\nabla f_t(x_t)\|^2}.$$

Adaptive Online Gradient Descent

A bit surprising result (verify it by yourself):

$$\sum_{t=1}^{T} \frac{\|\nabla f_t(x_t)\|^2}{\sqrt{\sum_{s=1}^{t} \|\nabla f_s(x_s)\|^2}} \le 2\sqrt{\sum_{t=1}^{T} \|\nabla f_t(x_t)\|^2}.$$

Finally, we achieve an adaptive regret without any future information:

$$\text{Regret}(T) \le \frac{3}{2}\sqrt{\sum_{t=1}^{T} \|\nabla f_t(x_t)\|^2}.$$

Hey, where is "smoothness" in online convex optimization?

Adaptive Online Gradient Descent

Recall a function is β-smoothness if for any $x, y \in \mathcal{K}$

$$f(y) - f(x) \le \langle y - x, \nabla f(x) \rangle + \frac{\beta}{2}\|y - x\|^2,$$

which implies

$$\|\nabla f(x)\|^2 \le 2\beta \left(f(x) - \min_{y \in \mathcal{K}} f(y) \right).$$

Assume functions $\{f_t\}$ are β-smoothness and non-negative, the regret again becomes "adaptive" to values of functions:

$$\text{Regret}(T) \le \frac{3}{2}\sqrt{2\beta \sum_{t=1}^{T} \left(f_t(x_t) - \min_{y \in \mathcal{K}} f_t(y) \right)} \le \frac{3}{2}\sqrt{2\beta \sum_{t=1}^{T} f_t(x_t)}.$$

Adaptive Online Gradient Descent

We have an interesting "self-bounds":

$$\sum_{t=1}^{T} f_t(x_t) - \min_{x\in\mathcal{K}} \sum_{t=1}^{T} f_t(x) \leq \frac{3}{2}\sqrt{2\beta \sum_{t=1}^{T} f_t(x_t)}.$$

It can read as

$$L_T - L^* \leq \sqrt{c \times L_T},$$

which implies (if $L_T, L^* \geq 0$)

$$L_T - L^* \leq c + 2\sqrt{c \times L^*}.$$

We have

$$\sum_{t=1}^{T} f_t(x_t) - \min_{x\in\mathcal{K}} \sum_{t=1}^{T} f_t(x) \leq \frac{9\beta}{2} + \sqrt{18\beta \min_{x\in\mathcal{K}} \sum_{t=1}^{T} f_t(x)}.$$

Adaptive Online Gradient Descent: AdaGrad

The regret is decomposed to be

$$
\begin{aligned}
\sum_{t=1}^{T} f_t(x_t) - \sum_{t=1}^{T} f_t(x) &\le \sum_{t=1}^{T} \langle x_t - x, \nabla f_t(x_t) \rangle \\
&= \sum_{i=1}^{d} \sum_{t=1}^{T} \langle x_{t,i} - x_i, \nabla f_{t,i}(x_t) \rangle \\
&= \sum_{i=1}^{d} \mathrm{Regret}_i(T)
\end{aligned}
$$

Recall $\eta_t = 1/\sqrt{\sum_{s=1}^{t} \|\nabla f_s(x_s)\|^2}$, can we use the adaptive gradient for each coordinate?

$$
\eta_{t,i} = \frac{1}{\sqrt{\sum_{s=1}^{t} \|\nabla f_{s,i}(x_s)\|^2}}.
$$

AdaGrad for Hyperrectangles

Initialization: each coordinate is in $[0, 1]$ and x_1.
For $t = 1, \cdots, T$:

- **Learner:** Submit x_t.
- **Environment:** Observe the loss $f_t(x)$.
- **Update for each coordinate:**
$$\eta_{t,i} = \frac{1}{\sqrt{\sum_{s=1}^{t} \|\nabla f_{s,i}(x_s)\|^2}}, \; x_{t+1,i} = x_{t,i} - \eta_{t,i} \nabla f_{t,i}(x_t).$$

AdaGrad[2] has key ingredients:

- A coordinate-wise learning process.
- The adaptive learning rates of $\{\eta_{t,i}\}$.

[2] J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. In COLT, 2010.

Adaptive Online Gradient Descent: AdaGrad

By using the gradient of η_t, the previous regret

$$\text{Regret}(T) \leq \frac{3}{2}\sqrt{d\sum_{t=1}^{T}\|\nabla f_t(x_t)\|^2}.$$

By using the gradient of $\eta_{t,i}$ for each coordinate, we have

$$\text{Regret}(T) \leq \frac{3}{2}\sum_{i=1}^{d}\sqrt{\sum_{t=1}^{T}\|\nabla f_{t,i}(x_t)\|^2}.$$

which one is better?

From AdaGrad to Adam

Adam for Stochastic Optimization

Initialization: γ_0 and γ_1 the discounted factors for the moment and learning rates; ϵ is the small constant.

For $t = 1, \cdots, T$:

- **Compute:**

$$\begin{aligned} m_t =& \gamma_0 m_{t-1} + (1-\gamma_0)\nabla f_t(x_t) \\ g_{t,i} =& \gamma_1 g_{t-1,i} + (1-\gamma_1)(\nabla f_{t,i}(x_t))^2 \end{aligned}$$

- **Bias-correcting:** $\hat{m}_t = m_t/(1-(\gamma_0)^t)$, $\hat{g}_{t,i} = g_{t,i}/(1-(\gamma_1)^t)$.
- **Update for each coordinate:**

$$\eta_{t,i} = \frac{1}{\sqrt{\hat{g}_{t,i}} + \epsilon}, \; x_{t+1,i} = x_{t,i} - \eta_{t,i}\hat{m}_{t,i}.$$

Adam is AdaGrad with "moment"!