

# Parameter-free version of Adaptive Gradient Methods for Strongly-Convex Functions

**Deepak Gouda**

*Georgia Institute of Technology  
Atlanta, GA, USA*

DEEPAKGODA@GATECH.EDU

**Hassan Naveed**

*Georgia Institute of Technology  
Atlanta, GA, USA*

HNAVEED3@GATECH.EDU

**Salil Kamath**

*Georgia Institute of Technology  
Atlanta, GA, USA*

SKAMATH36@GATECH.EDU

**Editors:** Jacob Abernethy and Shivani Agarwal

## Abstract

The optimal learning rate for adaptive gradient methods applied to  $\lambda$ -strongly convex functions relies on the parameters  $\lambda$  and learning rate  $\eta$ . In this paper, we adapt a universal algorithm along the lines of Metagrad, to get rid of this dependence on  $\lambda$  and  $\eta$ . The main idea is to concurrently run multiple experts and combine their predictions to a “master” algorithm. This master enjoys  $O(d \log T)$  regret bounds.

**Keywords:** Online Convex Optimization, Adagrad, Metagrad, Strongly-Convex functions

## 1. Introduction

Online convex optimization (OCO) is a widely recognized approach for representing sequential decision-making processes. The OCO algorithm operates in the following manner:

- the learner selects an action  $\mathbf{x}_t$
- an adversary chooses a loss function  $f_t(\cdot)$
- the learner incurs a loss  $f_t(\mathbf{x}_t)$
- the learner updates its action based on this loss

The aim is to minimize regret, which is defined as the disparity between the cumulative loss suffered by the learner and that of the optimal action in retrospect [Hazan \(2021\)](#). This is defined as:

$$Regret = \sum_{t=1}^T f_t(\mathbf{x}_t) - \min_{\mathbf{x} \in D} \sum_{t=1}^T f_t(\mathbf{x})$$

Several work has been done on this under various loss function settings. Among the most frequent are convex functions, exponentially concave functions and strongly convex functions. These leverage the slope of the function to alter the gradient update step in order to obtain sublinear regret

bounds. OGD on convex functions achieves  $O(\sqrt{T})$  regret with step size decreasing by  $1/\sqrt{T}$  [Zinkevich \(2003\)](#), and a  $O(\log(T))$  regret bound on strongly convex functions with a step-size of order  $1/T$  [Hazan et al. \(2006\)](#). Similarly, for exponentially concave functions, the ONS algorithm achieves a regret bound of  $O(\log(T))$  as well [Hazan et al. \(2006\)](#).

Our proposed algorithm combines a variant of the meta algorithm described in [Wang et al. \(2019\)](#) using strongly convex experts from a version of Adagrad described in [Duchi et al. \(2011\)](#).

## 2. Related Work

Recent work on adaptive gradients set a parameter specific learning rate to help with sparse gradients, i.e, settings where we only have a few non-zero features. A popular method was Adagrad, which decays the learning rate faster for parameters with larger gradients [Duchi et al. \(2011\)](#). This way, we have a larger step size for weights that have smaller magnitude. It still has a regret bound of the order  $O(\sqrt{T})$  similar to work by [Hazan \(2021\)](#), but this data dependent bound demonstrates better performance in practice.

The Adagrad algorithm without modification did not guarantee a  $O(\log(T))$  bound in strongly convex functions. This extension to the strongly convex case was made by [Mukkamala and Hein \(2017\)](#) by one major change. First, they decayed the step size on the parameters using the sum of squared gradients, rather than taking an L2 norm of them. They also had a decay rate for the constant added by Adagrad numerical stability, but this did not result in any better bounds or performance.

At another front, there has been a push to create universal algorithms that can perform well for most function types. As previously mentioned, the type of algorithm and learning rate strictly divided the type of functions into three main categories: convex, exp-concave, and strongly convex. It would be preferable to have algorithms that work for not only these three cases at once, but also everything in between. This influenced work on an algorithm called *MetaGrad* by [van Erven and Koolen \(2016\)](#), which works for functions that satisfy the so called Bernstein condition. It consists of one master algorithm which spawns several experts. Each of these slaves has its own learning rate and employs a surrogate loss function which closely mimics the Online Newton Step (ONS). The master algorithm combines the predictions from each of these slaves and passes down its gradients to them after incurring a loss. This approach only requires gradient information and achieves  $O(\sqrt{T \log T \log T})$  regret on convex and  $O(d \log T)$  on exponentially concave functions.

However, the  $O(d \log T)$  regret Metagrad implies on strongly convex functions has an  $O(d)$  gap from the optimal  $O(\log T)$  regret possible. This inspired work on *Maler* by [Wang et al. \(2019\)](#). The main idea was to run multiple learning algorithms for each of the three function classes in parallel. The master algorithm continued to track the best one on the fly. This spawned three times the number of experts as Metagrad, but enjoyed optimal regret bound on all three function classes. We borrow inspirations the learning algorithm employed by Maler on the strongly convex case and extend it to handle sparse gradients.

## 3. Setup

### 3.1. Notation

We denote  $C \subseteq \mathbb{R}^d$  to be our convex decision set. We let  $\mathbf{x}_t \in C$  be the learner's decision at time step  $t$ . We denote our loss function at time step  $t$  to be  $f_t(\cdot) : C \rightarrow \mathbb{R}$ . We let  $\mathbf{g}_t$  be the gradient of

$f_t(\mathbf{x}_t)$ . We use  $\|\cdot\|$  to denote the 2-norm, and  $\langle \cdot, \cdot \rangle$  to denote the Euclidean inner product. We use boldface letters to denote vectors.

### 3.2. Assumptions and Definitions

We include two main assumptions akin to those from previous studies [Wang et al. \(2019\)](#).

**Assumption 1:** The gradients of all loss functions  $f_1(\cdot) \dots f_T(\cdot)$  are bounded i.e.  $\forall t \in [T], \forall \mathbf{x} \in C$   $\|\nabla f_t(\mathbf{x})\| \leq G$  for some  $G \in \mathbb{R}$ .

**Assumption 2:** The diameter of the decision set  $C$  is bounded i.e.  $\forall \mathbf{x}_1, \mathbf{x}_2 \in C, \|\mathbf{x}_1 - \mathbf{x}_2\| \leq D$  for some  $D \in \mathbb{R}$ .

**Definition 1: Strong Convexity** A function  $f : C \rightarrow \mathbb{R}$  is  $\lambda$ -strongly convex if  $\forall \mathbf{x}_1, \mathbf{x}_2 \in C$ ,

$$f(\mathbf{x}_1) \geq f(\mathbf{x}_2) + \langle \nabla f(\mathbf{x}_2), \mathbf{x}_1 - \mathbf{x}_2 \rangle + \frac{\lambda}{2} \|\mathbf{x}_1 - \mathbf{x}_2\|^2$$

**Assumption 3:** All loss functions  $f_1(\cdot) \dots f_T(\cdot)$  are  $\lambda$ -strongly convex.

**Definition 2: A-weighted norm** Let  $A \in \mathbb{R}^{d \times d}$  and let  $\mathbf{x} \in \mathbb{R}^d$ . We define the  $A$ -weighted norm to be the matrix norm  $\|\mathbf{x}\|_A^2 = \mathbf{x}^\top A \mathbf{x} = \sum_{i,j=1}^d A_{ij} \mathbf{x}_i \mathbf{x}_j$  [Mukkamala and Hein \(2017\)](#).

**Definition 3: Weighted Projection** We define a weighted projection function onto the convex set  $C$  with respect to the  $A$ -weighted norm to be

$$\Pi_C^A(\mathbf{x}_1) = \arg \min_{\mathbf{x} \in C} \|\mathbf{x}_1 - \mathbf{x}\|_A^2$$

## 4. Algorithm

The algorithm draws inspiration from the strongly convex experts in Maler. It similarly follows a two hierarchical structure. At the lower level, we have a set of experts, each with their own learning rates  $\eta$ . Because we are only dealing with the strongly convex case, all of them use the same learning algorithm (Algorithm 2). At the higher level, we have a meta algorithm which tracks the best one using a tilted exponential weighted average [van Erven and Koolen \(2016\)](#).

---

### Algorithm 1 Meta Algorithm

---

**Input:** Grid of learning rates  $\eta_1, \eta_2, \dots, \eta_k$  and weights  $\pi_1^{\eta_1}, \pi_1^{\eta_2}, \dots, \pi_1^{\eta_k}$

Get prediction  $\mathbf{x}_t^\eta$  from Algorithm 2

$$\text{Play } \mathbf{x}_t = \frac{\sum_\eta \pi_t^\eta \eta \mathbf{x}_t^\eta}{\sum_\eta \pi_t^\eta \eta}$$

Observe gradient  $\mathbf{g}_t$  and send  $\mathbf{x}_t, \mathbf{g}_t$  to all experts

For all  $\eta$ , update weights according to

$$\pi_{t+1}^\eta = \frac{\pi_t^\eta \exp(-s_t^\eta(\mathbf{x}_t^\eta))}{\sum_\eta \pi_t^\eta \exp(-s_t^\eta(\mathbf{x}_t^\eta))}$$


---

We define the surrogate loss function to be

$$s_t^\eta(\mathbf{x}) = -\eta(\mathbf{x}_t - \mathbf{x})^\top \mathbf{g}_t + \eta^2 G^2 \|\mathbf{x}_t - \mathbf{x}\|^2 \quad (1)$$

Note, by taking the second derivative of the surrogate loss function, we see that it is  $\eta^2 G^2$  strongly convex.

---

**Algorithm 2** SC-Adagrad expert algorithm
 

---

**Input:**  $\mathbf{v}_0 = \mathbf{0} \in \mathbb{R}^d, \eta \in \mathbb{R}, \mathbf{x}_1^\eta = \mathbf{0}, \delta > 0$   
**for**  $t = 1 \dots T$  **do**  
     Send  $\mathbf{x}_t^\eta$  to Algorithm 1  
     Receive  $\mathbf{x}_t, \mathbf{g}_t$  from Algorithm 1  
      $\nabla s_t^\eta(\mathbf{x}_t^\eta) = \eta \mathbf{g}_t + 2\eta^2 G^2 (\mathbf{x}_t^\eta - \mathbf{x}_t)$   
      $\mathbf{v}_t = \mathbf{v}_{t-1} + \nabla s_t^\eta(\mathbf{x}_t^\eta) \odot \nabla s_t^\eta(\mathbf{x}_t^\eta)$   
      $A_t = \text{diag}(\mathbf{v}_t) + \delta I$   
      $\mathbf{x}_{t+1}^\eta = \Pi_C^{A_t} (\mathbf{x}_t^\eta - \alpha A_t^{-1} \nabla s_t^\eta(\mathbf{x}_t^\eta))$ , where  $\alpha = \frac{1}{4\eta^2}$   
**end**

---

**Experts:** Each expert uses an adaptive algorithm with an update rule derived from the Adagrad for strongly convex function [Mukkamala and Hein \(2017\)](#). However, in contrast to the original Adagrad, this adaptation is on the gradients of the surrogate losses, rather than the  $\mathbf{g}_t$  incurred by a single learner. We maintain  $\lceil \frac{1}{2} \log_2 T \rceil$  experts. For  $i = 0, 1, 2, \dots, \frac{1}{2} \log_2 T$ , each expert is configured with:

$$\eta_i = \frac{2^{-i}}{5GD}$$

and

$$\pi_1^{\eta_i} = \frac{c}{3(i+1)(i+2)}$$

These values for  $\eta$ ,  $\pi$  and  $\alpha$  were carefully chosen to satisfy certain criteria which will help in proving regret bounds.

## 5. Regret Analysis

### 5.1. Preliminaries

**Lemma 1** [Lemma 3.2 from [Mukkamala and Hein \(2017\)](#)] Let assumptions **A1** and **A2** hold, then for all  $T \geq 1$  and  $A_t, \delta_t$  defined from algorithm 2 and  $\nabla s_t = s_t^\eta(\mathbf{x}_t^\eta)$ , we have:

$$\sum_{t=1}^T \langle \nabla s_t, A_t^{-1} \nabla s_t \rangle \leq \sum_{i=1}^d \log \left( \frac{\|\nabla s_{1:T,i}\|^2 + \delta}{\delta} \right)$$

**Lemma 2**  $\forall t$ , let  $A_t$  be positive semi-definite diagonal matrices as which follows the update rules in Algorithm 2. Similarly, let  $\alpha \geq \frac{1}{4\eta^2}$  for some constant  $\eta$  and let  $\mu = 2\eta^2 G^2$ . Then the following holds:

$$A_t - A_{t-1} - 2\alpha \text{diag}(\mu) \preceq 0$$

and

$$A_1 - 2\alpha \text{diag}(\mu) \preceq \delta$$

**Proof:** In our update rule we have:

$$\begin{aligned}
 A_t &= \text{diag}(v_t) + \text{diag}(\delta) \\
 &= v_{t-1} + \nabla s_t^\eta(\mathbf{x}_t^\eta) \bullet \nabla s_t^\eta(\mathbf{x}_t^\eta) + \delta \\
 &= A_{t-1} + \nabla s_t^\eta(\mathbf{x}_t^\eta) \bullet \nabla s_t^\eta(\mathbf{x}_t^\eta) \\
 &\leq A_{t-1} + (G^s)^2
 \end{aligned}$$

where,  $\nabla s_t^\eta(\mathbf{x}_t^\eta) \bullet \nabla s_t^\eta(\mathbf{x}_t^\eta)$  has an upper bound of  $(G^s)^2$ .

This can be rearranged as:

$$\begin{aligned}
 A_t - A_{t-1} &\leq (G^s)^2 \\
 \Rightarrow A_t - A_{t-1} - 2\alpha \text{diag}(\mu) &\leq (G^s)^2 - 2\alpha \text{diag}(\mu) \\
 &\leq (G^s)^2 - 2 \left( \frac{1}{4\eta^2} \right) (2\eta^2 (G^s)^2) \\
 &\leq (G^s)^2 - (G^s)^2 = 0
 \end{aligned}$$

This proves our the first inequality. For the second one, since we have  $\mathbf{v}_0 = 0$  in our algorithm, we observe that  $A_1 = s_1^\eta(\mathbf{x}_1^\eta) + s_1^\eta(\mathbf{x}_1^\eta) + \text{diag}(\delta)$ . With the same setting of  $\alpha$ , we have:

$$\begin{aligned}
 A_1 &\leq (G^s)^2 + \delta \\
 A_1 - 2\alpha \text{diag}(\mu) &\leq (G^s)^2 - 2\alpha \text{diag}(\mu) + \delta \\
 &\leq \delta
 \end{aligned}$$

This proves the second inequality.

## 5.2. Meta Regret

The regret of the meta algorithm is defined as the difference of the surrogate losses of the meta learner and the surrogate losses of a given expert. It is given by:

$$R_T^{\text{Meta}} = \sum_{t=1}^T s_t(\mathbf{x}_t) - \sum_{t=1}^T s_t^\eta(\mathbf{x}_t^\eta) \quad (5.2.1)$$

**Lemma 3** [Lemma 1 in Wang et al. (2019)] For every grid point  $\eta$  we have:

$$\sum_{t=1}^T s_t(\mathbf{x}_t) - \sum_{t=1}^T s_t^\eta(\mathbf{x}_t^\eta) \leq 2 \ln \left( \sqrt{3} \left( \frac{1}{2} \log_2 T + 3 \right) \right)$$

The proof is attached in the appendix.

### 5.3. Expert Regret

The regret of the expert algorithm is defined as the difference of the surrogate losses of the expert and the surrogate losses of the best decision in hindsight. It is given by:

$$R_T^{Expert} = \sum_{t=1}^T s_t^\eta(\mathbf{x}_t^\eta) - \sum_{t=1}^T s_t^\eta(\mathbf{x}^*) \quad (5.3.1)$$

Let us denote  $\nabla s_t = \nabla s_t^\eta(\mathbf{x}^\eta)$

#### Lemma 4

$$(\mathbf{x}_t^\eta - \mathbf{x}^*)^\top \nabla s_t \leq \frac{\|\mathbf{x}_t^\eta - \mathbf{x}^*\|_{A_t}^2 - \|\mathbf{x}_{t+1}^\eta - \mathbf{x}^*\|_{A_t}^2}{2\alpha} + \frac{\alpha}{2} \langle \nabla s_t, A_t^{-1} \nabla s_t \rangle$$

#### Proof :

From the update rule, we have :

$$\begin{aligned} \|\mathbf{x}_{t+1}^\eta - \mathbf{x}^*\|_{A_t} &= \left\| \Pi_C^{A_t}(\mathbf{x}_t^\eta - \alpha A_t^{-1} \nabla s_t - \mathbf{x}^*) \right\|_{A_t}^2 \\ &\leq \left\| \mathbf{x}_t^\eta - \alpha A_t^{-1} \nabla s_t - \mathbf{x}^* \right\|_{A_t}^2 \\ &= \|\mathbf{x}_t^\eta - \mathbf{x}^*\|_{A_t}^2 + \alpha^2 \|A_t^{-1} \nabla s_t\|_{A_t}^2 - 2\alpha (\mathbf{x}_t^\eta - \mathbf{x}^*)^\top A_t (A_t^{-1} \nabla s_t) \\ \Rightarrow 2\alpha (\mathbf{x}_t^\eta - \mathbf{x}^*)^\top \nabla s_t &\leq \|\mathbf{x}_t^\eta - \mathbf{x}^*\|_{A_t}^2 - \|\mathbf{x}_{t+1}^\eta - \mathbf{x}^*\|_{A_t}^2 + \alpha^2 \|A_t^{-1} \nabla s_t\|_{A_t}^2 \\ \Rightarrow (\mathbf{x}_t^\eta - \mathbf{x}^*)^\top \nabla s_t &\leq \frac{\|\mathbf{x}_t^\eta - \mathbf{x}^*\|_{A_t}^2 - \|\mathbf{x}_{t+1}^\eta - \mathbf{x}^*\|_{A_t}^2}{2\alpha} + \frac{\alpha}{2} \langle \nabla s_t, A_t^{-1} \nabla s_t \rangle \end{aligned}$$

**Lemma 5** *The expert regret is given by:*

$$R_T^{Expert} = \sum_{t=1}^T s_t^\eta(\mathbf{x}_t^\eta) - s_t^\eta(\mathbf{x}^*) \leq \frac{D^2 d \delta}{2\alpha} + \frac{\alpha}{2} \sum_{i=1}^d \log \left( \frac{\|\nabla s_{1:T,i}\|^2 + \delta}{\delta} \right)$$

**Proof:** We can start from the definition of strong convexity

$$\begin{aligned}
 \sum_{t=1}^T s_t^\eta(\mathbf{x}_t^\eta) - s_t^\eta(\mathbf{x}^*) &\leq \sum_{t=1}^T \langle \nabla s_t^\eta(\mathbf{x}^\eta), \mathbf{x}_t - \mathbf{x}^* \rangle - \|\mathbf{x}_t^\eta - \mathbf{x}^*\|_{\text{diag}(\mu)}^2 \\
 &\leq \sum_{t=1}^T \frac{\|\mathbf{x}_t^\eta - \mathbf{x}^*\|_{A_t}^2 - \|\mathbf{x}_{t+1}^\eta - \mathbf{x}^*\|_{A_t}^2}{2\alpha} + \frac{\alpha}{2} \langle \nabla s_t, A_t^{-1} \nabla s_t \rangle - \|\mathbf{x}_t^\eta - \mathbf{x}^*\|_{\text{diag}(\mu)}^2 \\
 &\hspace{15em} \text{(Using Lemma 4)} \\
 &= \frac{\|\mathbf{x}_1^\eta - \mathbf{x}^*\|_{A_1}^2}{2\alpha} + \sum_{t=2}^T \frac{\|\mathbf{x}_t^\eta - \mathbf{x}^*\|_{A_t}^2 - \|\mathbf{x}_t^\eta - \mathbf{x}^*\|_{A_{t-1}}^2}{2\alpha} - \frac{\|\mathbf{x}_{T+1}^\eta - \mathbf{x}^*\|_{A_T}^2}{2\alpha} \\
 &\quad + \sum_{t=1}^T \frac{\alpha}{2} \langle \nabla s_t, A_t^{-1} \nabla s_t \rangle - \sum_{t=1}^T \|\mathbf{x}_t^\eta - \mathbf{x}^*\|_{\text{diag}(\mu)}^2 \\
 &\hspace{15em} \text{(Separating the first and last terms)} \\
 &\leq \frac{\|\mathbf{x}_1^\eta - \mathbf{x}^*\|_{A_1 - 2\alpha \text{diag}(\mu)}^2}{2\alpha} + \sum_{t=2}^T \frac{\|\mathbf{x}_t^\eta - \mathbf{x}^*\|_{A_t - A_{t-1} - 2\alpha \text{diag}(\mu)}^2}{2\alpha} \\
 &\quad + \frac{\alpha}{2} \langle \nabla s_t, A_t^{-1} \nabla s_t \rangle \\
 &\leq \frac{\|\mathbf{x}_1^\eta - \mathbf{x}^*\|_\delta^2}{2\alpha} + \sum_{t=2}^T \frac{\|\mathbf{x}_t^\eta - \mathbf{x}^*\|_0^2}{2\alpha} + \frac{\alpha}{2} \langle \nabla s_t, A_t^{-1} \nabla s_t \rangle \quad \text{(Using Lemma 2)} \\
 &\leq \frac{\|\mathbf{x}_1^\eta - \mathbf{x}^*\|_\delta^2}{2\alpha} + \frac{\alpha}{2} \langle \nabla s_t, A_t^{-1} \nabla s_t \rangle \quad \text{(because second term} \leq 0) \\
 &\leq \frac{D^2 d \delta}{2\alpha} + \frac{\alpha}{2} \langle \nabla s_t, A_t^{-1} \nabla s_t \rangle \quad \text{(Using A1)} \\
 &\leq \frac{D^2 d \delta}{2\alpha} + \frac{\alpha}{2} \sum_{i=1}^d \log \left( \frac{\|\nabla s_{1:T,i}\|^2 + \delta}{\delta} \right) \quad \text{(Using Lemma 1)}
 \end{aligned}$$

#### 5.4. Total Regret

The total regret is given by

$$\begin{aligned}
 R_T &= \sum_{t=1}^T f_t(\mathbf{x}_t) - f_t(\mathbf{x}^*) \\
 &\leq \sum_{t=1}^T \mathbf{g}_t^\top (\mathbf{x}_t - \mathbf{x}^*) - \frac{\mu}{2} \|\mathbf{x}_t - \mathbf{x}^*\|^2 \quad (\text{from definition of strongly convex functions}) \\
 &= \sum_{t=1}^T \frac{-s_t^\eta(\mathbf{x}^*) + \eta^2 G^2 \|\mathbf{x}_t - \mathbf{x}^*\|}{\eta} - \frac{\mu}{2} \|\mathbf{x}_t - \mathbf{x}^*\|^2 \\
 &\quad (\text{from the definition of surrogate loss function}) \\
 &\leq \sum_{t=1}^T \frac{s_t^\eta(\mathbf{x}_t) - s_t^\eta(\mathbf{x}^*)}{\eta} + \eta \sum_{t=1}^T G^2 \|\mathbf{x}_t - \mathbf{x}^*\| - \frac{\mu}{2} \sum_{t=1}^T \|\mathbf{x}_t - \mathbf{x}^*\|^2 \\
 &= \sum_{t=1}^T \frac{s_t^\eta(\mathbf{x}_t) - s_t^\eta(\mathbf{x}_t^\eta)}{\eta} + \sum_{t=1}^T \frac{s_t^\eta(\mathbf{x}_t^\eta) - s_t^\eta(\mathbf{x}^*)}{\eta} + \eta \sum_{t=1}^T G^2 \|\mathbf{x}_t - \mathbf{x}^*\| - \frac{\mu}{2} \sum_{t=1}^T \|\mathbf{x}_t - \mathbf{x}^*\|^2 \\
 &\leq \frac{2}{\eta} \log \left( \sqrt{3} \left( \frac{1}{2} \log(T) + 3 \right) \right) + \frac{1}{\eta} \frac{D^2 d \delta}{2\alpha} + \frac{\alpha}{2} \sum_{i=1}^d \log \left( \frac{\|\nabla s_{1:T,i}\|^2 + \delta}{\delta} \right) \\
 &\quad (\text{Using lemma 3, 5}) \\
 &\quad + \eta V_T^S - \frac{\mu}{2} \sum_{t=1}^T \|\mathbf{x}_t - \mathbf{x}^*\|^2 \tag{5.4.1}
 \end{aligned}$$

where,  $V_T^S = \sum_{t=1}^T G^2 \|\mathbf{x}_t - \mathbf{x}^*\|^2$ .

Let us denote

$$E_T = \frac{D^2 d \delta}{2\alpha} + \frac{\alpha}{2} \sum_{i=1}^d \log \left( \frac{\|\nabla s_{1:T,i}\|^2 + \delta}{\delta} \right) = O \left( d \log \left( \frac{T}{\delta} \right) \right) \tag{5.4.2}$$

$$A_T = 2 \log \left( \sqrt{3} \left( \frac{1}{2} \log(T) + 3 \right) \right) + E_T \tag{5.4.2}$$

Using (5.4.2) in (5.4.1), we have

$$R_T \leq \frac{A_T}{\eta} + \eta V_T^S - \frac{\mu}{2} \sum_{t=1}^T \|\mathbf{x}_t - \mathbf{x}^*\|^2 \tag{5.4.3}$$

The optimal  $\hat{\eta}$  to minimize  $A_T + \eta V_T^S$  is (Wang et al. (2019))

$$\hat{\eta} = \sqrt{\frac{A_T}{V_T^S}} \geq \frac{1}{5GD} \tag{5.4.4}$$



If  $\hat{\eta} \leq \frac{1}{5GD}$ , then by construction there exists a grid point  $\eta \in [\frac{\hat{\eta}}{2}, \hat{\eta}]$ . Thus.

$$\begin{aligned}
 R_T &\leq \eta V_T^S + \frac{A_T}{\eta} - \frac{\mu}{2} \sum_{t=1}^T \|\mathbf{x}_t - \mathbf{x}^*\|^2 \\
 &\leq \hat{\eta} V_T^S + \frac{2A_T}{\hat{\eta}} - \frac{\mu}{2} \sum_{t=1}^T \|\mathbf{x}_t - \mathbf{x}^*\|^2 \\
 &= 3\sqrt{V_T^S A_T} - \frac{\mu}{2} \sum_{t=1}^T \|\mathbf{x}_t - \mathbf{x}^*\|^2
 \end{aligned} \tag{5.4.5}$$

If  $\hat{\eta} \geq \frac{1}{5GD}$ , by (5.4.4), we have

$$V_T^S \leq 25G^2 D^2 A_T \tag{5.4.6}$$

Using (5.4.4), (5.4.6) and  $\hat{\eta} = \frac{1}{5GD}$  in (5.4.3), we have

$$\begin{aligned}
 R_T &\leq 2\sqrt{A_T V_T^S} - \frac{\mu}{2} \sum_{t=1}^T \|\mathbf{x}_t - \mathbf{x}^*\|^2 \\
 &\leq 10GDA_T - \frac{\mu}{2} \sum_{t=1}^T \|\mathbf{x}_t - \mathbf{x}^*\|^2
 \end{aligned} \tag{5.4.7}$$

Hence, combining (5.4.5) and (5.4.6), we have

$$R_T \leq 3\sqrt{V_T^S A_T} + 10GDA_T - \frac{\mu}{2} \sum_{t=1}^T \|\mathbf{x}_t - \mathbf{x}^*\|^2 \tag{5.4.8}$$

Let  $x = V_T^S$  and  $y = A_T$ . For  $x, y, \gamma \geq 0$

$$\begin{aligned}
 \sqrt{xy} &\leq \frac{\gamma}{2}x + \frac{1}{2\gamma}y \\
 \Rightarrow 3\sqrt{V_T^S A_T} &\leq \frac{3\gamma}{2}V_T^S + \frac{3}{2\gamma}A_T
 \end{aligned} \tag{5.4.9}$$

Using (5.4.9) in (5.4.8), we have

$$\begin{aligned}
 R_T &\leq \frac{3\gamma}{2} V_T^S + \frac{3}{2\gamma} A_T + 10GDA_T - \frac{\mu}{2} \sum_{t=1}^T \|x_t - x^*\|^2 \\
 &= \frac{3\gamma}{2} V_T^S + \frac{3}{2\gamma} A_T + 10GDA_T - \frac{\mu}{2G^2} V_T^S \\
 &= \left( \frac{3\gamma}{2} - \frac{\mu}{2G^2} \right) V_T^S + \left( \frac{3}{2\gamma} + 10GD \right) A_T
 \end{aligned}$$

Setting  $\gamma = \frac{\mu}{3G^2}$ , we have

$$\begin{aligned}
 R_T &= \left( \frac{G^2}{\mu} + 10GD \right) A_T \\
 &= \left( \frac{G^2}{\mu} + 10GD \right) \left( 2 \log \left( \sqrt{3} \left( \frac{1}{2} \log(T) + 3 \right) \right) + \frac{D^2 d \delta}{2\alpha} + \frac{\alpha}{2} \sum_{i=1}^d \log \left( \frac{\|\nabla s_{1:T,i}\|^2 + \delta}{\delta} \right) \right)
 \end{aligned}$$

**Remark :** In [Mukkamala and Hein \(2017\)](#) the authors claim that

$$\sum_{i=1}^d \log \left( \frac{\|\nabla s_{1:T,i}\|^2 + \delta_{T,i}}{\delta_{1,i}} \right) = O(\log T)$$

And hence, the overall regret is  $O(\log T)$  for strongly-convex functions. Even with the monotonic-decay factor, the term

$$\frac{\|\nabla s_{1:T,i}\|^2}{\delta_{1,i}} \leq \frac{TG^2}{\delta_{1,i}}$$

Hence, the regret term is bound by  $O(d \log(T))$  rather than  $O(\log(T))$ . This result hasn't been explained in the paper which seems like a weakness on the results. Hence, we take state the final regret bound as  $O(d \log(T))$ .

## 6. Conclusion and Future Work

By taking a Metagrad like approach to Adagrad, we can have created a parameter free version of Strongly Convex Adagrad. This enjoys  $O(d \log T)$  regret bound in strongly convex functions. If every expert employs SAdam [Wang et al. \(2020\)](#), the regret bound can be improved to  $O(\log T)$  instead. This work can be extended to make parameter free versions of other algorithms such as RMSProp or Adam.

## Acknowledgments

We would like to thank Jacob Abernethy, Tyler Labonte, Guanghui Wang, Zihao Hu and Yeojoon Youn for teaching this course and preparing us for this task. The guidance for this idea comes from Guanghui Wang.

## References

- John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(61):2121–2159, 2011. URL <http://jmlr.org/papers/v12/duchi11a.html>.
- Elad Hazan. Introduction to online convex optimization, 2021.
- Elad Hazan, Adam Kalai, Satyen Kale, and Amit Agarwal. Logarithmic regret algorithms for online convex optimization. pages 499–513, 06 2006. ISBN 978-3-540-35294-5. doi: 10.1007/11776420\_37.
- Mahesh Chandra Muckkamala and Matthias Hein. Variants of rmsprop and adagrad with logarithmic regret bounds, 2017.
- Tim van Erven and Wouter M. Koolen. Metagrad: Multiple learning rates in online learning, 2016.
- Guanghui Wang, Shiyin Lu, and Lijun Zhang. Adaptivity and optimality: A universal algorithm for online convex optimization, 2019.
- Guanghui Wang, Shiyin Lu, Quan Cheng, Wei wei Tu, and Lijun Zhang. Sadam: A variant of adam for strongly convex functions. In *International Conference on Learning Representations*, 2020. URL <https://openreview.net/forum?id=rye5YaEtPr>.
- Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the Twentieth International Conference on International Conference on Machine Learning*, ICML’03, page 928–935. AAAI Press, 2003. ISBN 1577351894.

## Appendix A. Proof of Meta Regret

This proof follows closely from the proof of the meta regret bound in Wang et al. (2019). We define a potential function

$$\Phi(T) = \sum_{\eta} \pi_1^{\eta} \exp \left( - \sum_{t=1}^T s_t^{\eta}(\mathbf{x}_t^{\eta}) \right)$$

We have

$$\begin{aligned} \Phi(T+1) - \Phi(T) &= \sum_{\eta} \pi_1^{\eta} \exp \left( - \sum_{t=1}^T s_t^{\eta}(\mathbf{x}_t^{\eta}) \right) \exp \left( -s_{T+1}^{\eta}(\mathbf{x}_{T+1}^{\eta}) - 1 \right) \\ &\leq \sum_{\eta} \pi_1^{\eta} \exp \left( - \sum_{t=1}^T s_t^{\eta}(\mathbf{x}_t^{\eta}) \right) \eta(\mathbf{x}_{T+1} - \mathbf{x}_{T+1}^{\eta})^{\top} \mathbf{g}_t \\ &= (\mathbf{a}_T \mathbf{x}_{T+1} - \mathbf{b}_T)^{\top} \mathbf{g}_t \end{aligned}$$

where

$$\begin{aligned}\mathbf{a}_T &= \sum_{\eta} \pi_1^{\eta} \exp \left( - \sum_{t=1}^T s_t^{\eta}(\mathbf{x}_t^{\eta}) \right) \eta \\ \mathbf{b}_T &= \sum_{\eta} \pi_1^{\eta} \exp \left( - \sum_{t=1}^T s_t^{\eta}(\mathbf{x}_t^{\eta}) \right) \eta \mathbf{x}_{T+1}^{\eta}\end{aligned}$$

By definition,

$$\begin{aligned}\mathbf{x}_{T+1} &= \frac{\sum_{\eta} \pi_{T+1}^{\eta} \eta \mathbf{x}_{T+1}^{\eta}}{\sum_{\eta} \pi_{T+1}^{\eta} \eta} \\ &= \frac{\sum_{\eta} \pi_1^{\eta} \exp \left( - \sum_{t=1}^{T+1} s_t^{\eta}(\mathbf{x}_t^{\eta}) \right) \eta \mathbf{x}_T^{\eta}}{\sum_{\eta} \pi_1^{\eta} \exp \left( - \sum_{t=1}^T s_t^{\eta}(\mathbf{x}_t^{\eta}) \right) \eta} \\ &= \frac{\mathbf{b}_T}{\mathbf{a}_T}\end{aligned}$$

This implies that  $\Phi(T+1) - \Phi(T) \leq 0$  and  $1 = \Phi(0) \geq \Phi(1) \geq \dots \Phi(T) \geq 0$ . Then for any  $\eta$

$$\begin{aligned}\ln \left( \pi_1^{\eta} \exp \left( - \sum_{t=1}^T s_t^{\eta}(\mathbf{x}_t^{\eta}) \right) \right) &\leq 0 \\ \implies \ln \left( \frac{1}{\pi_1^{\eta}} \right) + \sum_{t=1}^T s_t^{\eta}(\mathbf{x}_t^{\eta}) &\geq 0\end{aligned}$$

By definition,  $\pi_1^{\eta_i} = \frac{c}{3(i+1)(i+2)}$ , where  $c = 1 + \frac{1}{1+k}$ ,  $i \in \{0, \dots, k\}$  and  $k = \lceil \frac{1}{2} \log_2(T) \rceil$ . Therefore,

$$\begin{aligned}\ln \left( \frac{1}{\pi_1^{\eta}} \right) &= \ln \left( \frac{3(i+1)(i+2)}{c} \right) \\ &\leq \ln (3(1+k)(2+k)) \\ &\leq \ln \left( \left( \sqrt{3} \left( 3 + \frac{1}{2} \log_2(T) \right) \right)^2 \right) \\ &= 2 \ln \left( \sqrt{3} \left( 3 + \frac{1}{2} \log_2(T) \right) \right)\end{aligned}$$