Online Convex Optimization
Independent Study Report

M Elamparithy

December 2023

Contents

1 Introduction

The following algorithms and theorems are an exploration into online learning and optimization. We first present the basics of Convex optimization and online learning from [1].

2 Basics of convex optimization

A function $f : \mathcal{K} \mapsto \mathbb{R}$ is convex if for any $\mathbf{x}, \mathbf{y} \in \mathcal{K}$

$$\forall \alpha \in [0,1], \; f((1-\alpha)\mathbf{x} + \alpha\mathbf{y}) \leq (1-\alpha)f(\mathbf{x}) + \alpha f(\mathbf{y}).$$

If f is differentiable, then it is convex if and only if $\forall \mathbf{x}, \mathbf{y} \in \mathcal{K}$

$$f(\mathbf{y}) \geq f(\mathbf{x}) + \nabla f(\mathbf{x})^\top (\mathbf{y} - \mathbf{x}).$$

For convex and non-differentiable functions f, the subgradient at $\mathbf{x}$ is *defined* to be any member of the set of vectors $\{\nabla f(\mathbf{x})\}$ that satisfies the above for all $\mathbf{y} \in \mathcal{K}$.

A f is Lipschitz continuous with parameter G if, for all $\mathbf{x}, \mathbf{y} \in \mathcal{K}$

$$|f(\mathbf{x}) - f(\mathbf{y})| \leq G\|\mathbf{x} - \mathbf{y}\|.$$

A function is α-strongly convex if

$$f(\mathbf{y}) \geq f(\mathbf{x}) + \nabla f(\mathbf{x})^\top (\mathbf{y} - \mathbf{x}) + \frac{\alpha}{2}\|\mathbf{y} - \mathbf{x}\|^2.$$

A function is β-smooth if

$$f(\mathbf{y}) \leq f(\mathbf{x}) + \nabla f(\mathbf{x})^\top (\mathbf{y} - \mathbf{x}) + \frac{\beta}{2}\|\mathbf{y} - \mathbf{x}\|^2.$$

The latter condition is equivalent to a Lipschitz condition over the gradients, i.e.,

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

If the function is twice differentiable and admits a second derivative, then the above conditions are equivalent to the following condition on the Hessian, denoted $\nabla^2 f(\mathbf{x})$:

$$\alpha I \preccurlyeq \nabla^2 f(\mathbf{x}) \preccurlyeq \beta I,$$

where $A \preccurlyeq B$ if the matrix $B - A$ is positive semidefinite.

When the function f is both α-strongly convex and β-smooth, it is said to be γ-well-conditioned where γ is called the *condition number* of f

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

Algorithm 1 Gradient Descent

```
Input: time horizon $T$, initial point $x_0$, step sizes $\{\eta_t\}$
for $t = 0, \dots, T-1$ do
    $\mathbf{x}_{t+1} = \mathbf{x}_t - \eta_t \nabla_t$
end for
return $\bar{\mathbf{x}} = \arg\min_{\mathbf{x}_t} \{f(\mathbf{x}_t)\}$
```

2.1 Gradient Descent

Let,

1. Distance to optimality in value: $h_t = h(\mathbf{x}_t) = f(\mathbf{x}_t) - f(\mathbf{x}^*)$
2. Euclidean distance to optimality: $d_t = \|\mathbf{x}_t - \mathbf{x}^*\|$
3. Current gradient norm $\|\nabla_t\| = \|\nabla f(\mathbf{x}_t)\|$

With these notations we define the Gradient Descent with Polyak step size algorithm-

Algorithm 2 Gradient Descent with Polyak stepsize

```
Input: time horizon $T$, $x_0$
for $t = 0, \dots, T-1$ do
    Set $\eta_t = \frac{h_t}{\|\nabla_t\|^2}$
    $\mathbf{x}_{t+1} = \mathbf{x}_t - \eta_t \nabla_t$
end for
Return $\bar{\mathbf{x}} = \arg\min_{\mathbf{x}_t} \{f(\mathbf{x}_t)\}$
```

We assume $\|\nabla_t\| \leq G$, and define:

$$B_T \quad = \quad \min \left\{ \frac{G d_0}{\sqrt{T}}, \frac{2\beta d_0^2}{T}, \frac{3G^2}{\alpha T}, \beta d_0^2 \left(1 - \frac{\gamma}{4}\right)^T \right\}$$

We can now state the main guarantee of GD with the Polyak stepsize:

Theorem .1. *(GD with the Polyak Step Size) Algorithm 2 guarantees the following after T steps:*

$$f(\bar{\mathbf{x}}) - f(\mathbf{x}^\star) \leq \min_{0 \leq t \leq T} \{h_t\} \leq B_T$$

2.2 Constrained Gradient Descent

Theorem .2. *For constrained minimization of γ-well-conditioned functions and $\eta_t = \frac{1}{\beta}$, Algorithm 3 converges as*

$$h_{t+1} \leq h_1 \cdot e^{-\frac{\gamma t}{4}}$$

Algorithm 3 Constrained gradient descent

```
Input: $f$, $T$, initial point $\mathbf{x}_1 \in \mathcal{K}$, sequence of step sizes $\{\eta_t\}$
for $t = 1$ to $T$ do
    Let $\mathbf{y}_{t+1} = \mathbf{x}_t - \eta_t \nabla f(\mathbf{x}_t),\ \mathbf{x}_{t+1} = \Pi_{\mathcal{K}}(\mathbf{y}_{t+1})$
end for
return $\mathbf{x}_{T+1}$
```

2.3 Reductions to Non-Smooth and Non-Strongly convex functions

Algorithm 4 Gradient descent, reduction to β-smooth functions

```
Input: $f$, $T$, $\mathbf{x}_1 \in \mathcal{K}$, parameter $\tilde{\alpha}$.
Let $g(\mathbf{x}) = f(\mathbf{x}) + \frac{\tilde{\alpha}}{2}\|\mathbf{x} - \mathbf{x}_1\|^2$
Apply Algorithm 3 with parameters $g, T, \{\eta_t = \frac{1}{\beta}\}, \mathbf{x}_1$, return $\mathbf{x}_T$.
```

Theorem .3. *For β-smooth convex functions, Algorithm 4 with parameter $\tilde{\alpha} = \frac{\beta \log t}{D^2 t}$ converges as*

$$h_{t+1} = O\left(\frac{\beta \log t}{t}\right)$$

Algorithm 5 Gradient descent, reduction to non-smooth functions

```
Input: $f, \mathbf{x}_1, T, \delta$
Let $\hat{f}_\delta(\mathbf{x}) = \mathbf{E}_{\mathbf{v}\sim\mathbb{B}}[f(\mathbf{x} + \delta\mathbf{v})]$
Apply Algorithm 3 on $\hat{f}_\delta, \mathbf{x}_1, T, \{\eta_t = \delta\}$, return $\mathbf{x}_T$
```

Theorem .4. *For $\delta = \frac{dG}{\alpha}\frac{\log t}{t}$ Algorithm 5 converges as*

$$h_t = O\left(\frac{G^2 d \log t}{\alpha t}\right).$$

3 Online convex optimization: The basics

Consider a convex function f. In the online setting, we receive one point at a time to make our predictions. We now define a metric called *Regret* which is appropriate for the online setting.

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

3.1 Online Gradient Descent

Algorithm 6 online gradient descent

```
Input: convex set $\mathcal{K}$, $T$, $\mathbf{x}_1 \in \mathcal{K}$, step sizes $\{\eta_t\}$
for $t = 1$ to $T$ do
    Play $\mathbf{x}_t$ and observe cost $f_t(\mathbf{x}_t)$.
    Update and project:
        $\mathbf{y}_{t+1} = \mathbf{x}_t - \eta_t \nabla f_t(\mathbf{x}_t)$
        $\mathbf{x}_{t+1} = \prod_{\mathcal{K}}(\mathbf{y}_{t+1})$
end for
```

Theorem .5. *Given a convex function f, online gradient descent with step sizes $\{\eta_t = \frac{D}{G\sqrt{t}},\ t \in [T]\}$ guarantees the following for all $T \geq 1$:*

$$\mathrm{Regret}_T = \sum_{t=1}^{T} f_t(\mathbf{x}_t) - \min_{\mathbf{x}^\star\in\mathcal{K}} \sum_{t=1}^{T} f_t(\mathbf{x}^\star) \ \leq \frac{3}{2} GD\sqrt{T}.$$

Theorem .6. *For α-strongly convex loss functions, online gradient descent with step sizes $\eta_t = \frac{1}{\alpha t}$ achieves the following guarantee for all $T \geq 1$*

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

4 Second order algorithms

A convex function $f : \mathbb{R}^n \mapsto \mathbb{R}$ is defined to be α-exp-concave over $\mathcal{K} \subseteq \mathbb{R}^n$ if the function g is concave, where $g : \mathcal{K} \mapsto \mathbb{R}$ is defined as

$$g(\mathbf{x}) = e^{-\alpha f(\mathbf{x})}$$

Exp-concavity implies strong-convexity in the direction of the gradient-:

Lemma .7. *A twice-differentiable function $f : \mathbb{R}^n \mapsto \mathbb{R}$ is α-exp-concave at $\mathbf{x}$ if and only if*

$$\nabla^2 f(\mathbf{x}) \succcurlyeq \alpha \nabla f(\mathbf{x}) \nabla f(\mathbf{x})^\top.$$

Lemma .8. *Let $f : \mathcal{K} \to \mathbb{R}$ be an α-exp-concave function, and D, G denote the diameter of $\mathcal{K}$ and a bound on the (sub)gradients of f respectively. The following holds for all $\gamma \leq \frac{1}{2} \min\{\frac{1}{GD}, \alpha\}$ and all $\mathbf{x}, \mathbf{y} \in \mathcal{K}$:*

$$f(\mathbf{x}) \geq f(\mathbf{y}) + \nabla f(\mathbf{y})^\top (\mathbf{x} - \mathbf{y}) + \frac{\gamma}{2} (\mathbf{x} - \mathbf{y})^\top \nabla f(\mathbf{y}) \nabla f(\mathbf{y})^\top (\mathbf{x} - \mathbf{y}).$$

4.1 Exponentially Weighted Online Convex Optimization Algorithm

Algorithm 7 Exponentially Weighted Online Optimizer

```
1: Input: convex set $\mathcal{K}$, $T$, parameter $\alpha > 0$.
2: for $t = 1$ to $T$ do
3:     Let $w_t(\mathbf{x}) = e^{-\alpha \sum_{\tau=1}^{t-1} f_\tau(\mathbf{x})}$.
4:     Play $\mathbf{x}_t$ given by
           $$\mathbf{x}_t = \frac{\int_{\mathcal{K}} \mathbf{x} \; w_t(\mathbf{x}) d\mathbf{x}}{\int_{\mathcal{K}} w_t(\mathbf{x}) d\mathbf{x}}.$$
5: end for
```

Theorem .9. *Algorithm 7 achieves the following guarantee, for all $T \geq 1$.*

$$\mathrm{Regret}_T(EWOO) \;\leq\; \frac{d}{\alpha} \log T + \frac{2}{\alpha}.$$

Now, computing x_k *(step 4)* requires evaluating an integral which may be intractable. So we introduce another algorithm-

4.2 Online Newton Step Algorithm

Theorem .10. *Algorithm 8 with parameters $\gamma = \frac{1}{2} \min\{\frac{1}{GD}, \alpha\}$, $\varepsilon = \frac{1}{\gamma^2 D^2}$ and $T \geq 4$ guarantees*

$$\mathrm{Regret}_T \;\leq\; 2 \left(\frac{1}{\alpha} + GD \right) n \log T.$$

Algorithm 8 Online Newton Step

```
Input: convex set $\mathcal{K}$, $T$, $\mathbf{x}_1 \in \mathcal{K} \subseteq \mathbb{R}^n$, parameters $\gamma, \varepsilon > 0$, $A_0 = \varepsilon \mathbf{I}_n$
for $t = 1$ to $T$ do
    Play $\mathbf{x}_t$ and observe cost $f_t(\mathbf{x}_t)$.
    Rank-1 update: $A_t = A_{t-1} + \nabla_t \nabla_t^\top$
    Newton step and generalized projection:
        $$\mathbf{y}_{t+1} = \mathbf{x}_t - \frac{1}{\gamma} A_t^{-1} \nabla_t$$
        $$\mathbf{x}_{t+1} = \prod_{\mathcal{K}}^{A_t}(\mathbf{y}_{t+1}) = \arg\min_{\mathbf{x} \in \mathcal{K}} \left\{ \|\mathbf{y}_{t+1} - \mathbf{x}\|_{A_t}^2 \right\}$$
end for
```

5 Regularization

Regularization functions are strongly convex and smooth functions denoted by $R : \mathcal{K} \mapsto \mathbb{R}$.

The Bregman divergence ($B_R(\mathbf{x}||\mathbf{y})$) with respect to the function R, is defined as

$$B_R(\mathbf{x}||\mathbf{y}) = R(\mathbf{x}) - R(\mathbf{y}) - \nabla R(\mathbf{y})^\top (\mathbf{x} - \mathbf{y}).$$

5.1 Follow-The-Leader

Follow-The-Leader refers to the use of the optimal decision in hindsight. *i.e*

$$\mathbf{x}_{t+1} = \arg\min_{\mathbf{x} \in \mathcal{K}} \sum_{\tau=1}^{t} f_\tau(\mathbf{x}).$$

Follow-The-Leader can fail in simple settings because it is unstable. Thus, a modified version called the Regularized-Follow-The-Leader is introduced.

Algorithm 9 Regularized Follow The Leader

```
Input: $\eta > 0$, regularization function $R$, and a bounded, convex and closed set $\mathcal{K}$.
Let $\mathbf{x}_1 = \arg\min_{\mathbf{x} \in \mathcal{K}} \{R(\mathbf{x})\}$.
for $t = 1$ to $T$ do
    Play $\mathbf{x}_t$ and observe cost $f_t(\mathbf{x}_t)$.
    Update
        $$\mathbf{x}_{t+1} = \arg\min_{\mathbf{x} \in \mathcal{K}} \left\{ \eta \sum_{s=1}^{t} \nabla_s^\top \mathbf{x} + R(\mathbf{x}) \right\}$$
end for
```

Theorem .11. *Given a convex function f, The RFTL Algorithm 9 attains for every $\mathbf{u} \in \mathcal{K}$ the following bound on the regret:*

$$\text{Regret}_T \leq 2\eta \sum_{t=1}^{T} \|\nabla_t\|_t^{*2} + \frac{R(\mathbf{u}) - R(\mathbf{x}_1)}{\eta}.$$

5.2 Online Mirror Descent

Online Mirror descent is the online version of a general class of first order methods generalizing gradient descent. OMD generalizes gradient descent by having the update being carried out in a "dual" space, where the duality notion is defined by the choice of regularization. The regularization transforms the space in which gradient updates are performed. This transformation enables better bounds in terms of the geometry of the space.

Algorithm 10 Online Mirror Descent

```
Input: parameter $\eta > 0$, regularization function $R(\mathbf{x})$.
Let $\mathbf{y}_1$ be such that $\nabla R(\mathbf{y}_1) = \mathbf{0}$ and $\mathbf{x}_1 = \arg\min_{\mathbf{x} \in \mathcal{K}} B_R(\mathbf{x} || \mathbf{y}_1)$.
for $t = 1$ to $T$ do
    Play $\mathbf{x}_t$.
    Observe the loss function $f_t$ and let $\nabla_t = \nabla f_t(\mathbf{x}_t)$.
    Update $\mathbf{y}_t$ according to the rule:
        [Lazy version]     $\nabla R(\mathbf{y}_{t+1}) = \nabla R(\mathbf{y}_t) - \eta \nabla_t$
        [Agile version]    $\nabla R(\mathbf{y}_{t+1}) = \nabla R(\mathbf{x}_t) - \eta \nabla_t$
    Project according to $B_R$:
        $\mathbf{x}_{t+1} = \arg\min_{\mathbf{x} \in \mathcal{K}} B_R(\mathbf{x} || \mathbf{y}_{t+1})$
end for
```

Theorem .12. *The OMD Algorithm 10 attains for every $\mathbf{u} \in \mathcal{K}$ the following bound on the regret:*

$$\text{Regret}_T \leq \frac{\eta}{4} \sum_{t=1}^{T} \|\nabla_t\|_t^{*2} + \frac{R(\mathbf{u}) - R(\mathbf{x}_1)}{2\eta}.$$

6 Accelerated Gradient methods

Given a convex function f, Accelerated Gradient Methods improve upon the simple gradient descent method-

$$x_{k+1} = x_k - \alpha \nabla f(x_k),$$

We now look at some accelerated methods presented in [2]

6.1 Momentum Method

The update step is defined as

$$x_{k+1} = x_k - \alpha \nabla f(x_k) + \beta \left(x_k - x_{k-1} \right), \tag{1}$$

Where $\alpha > 0$ and $\beta > 0$

Theorem .13. *Given an $L-$ smooth function f, the sequence $\{x_k\}$ generated by* (1) *satisfies*

$$\min_{0 \le k \le T} f(x_k) - f^\star \le \mathcal{O}\bigg(\frac{\|x_0 - x^\star\|^2}{T} \bigg),$$

for all $T \in \mathbb{N}_0$. In addition to that, for any fixed $\bar{\alpha} \in (0, 1/L]$ and $\beta = 1 - \sqrt{\bar{\alpha} L}$ the convergence factor is

$$\min_{0 \le k \le T} f(x_k) - f^\star \le \frac{1}{2(T+1)} \bigg(\frac{2\sqrt{\bar{\alpha} L} - \bar{\alpha} L}{\bar{\alpha}} \bigg) \|x_0 - x^\star\|^2.$$

Theorem .14. *Given a function f which is L-Smooth, μ-strongly convex and that*

$$\alpha \in (0, \frac{2}{L}), \quad 0 \le \beta < \frac{1}{2} \bigg(\frac{\mu \alpha}{2} + \sqrt{\frac{\mu^2 \alpha^2}{4} + 4(1 - \frac{\alpha L}{2})} \bigg).$$

Then, momentum method (1) *converges linearly to a unique optimizer $x^\star$. In particular,*

$$f(x_k) - f^\star \le q^k (f(x_0) - f^\star),$$

where $q \in [0, 1)$.

6.2 Nesterov's Method

The update step is defined as

$$\begin{aligned} y_{k+1} &= x_k - \frac{1}{L} \nabla f(x_k), \\ x_{k+1} &= (1 - \gamma_k) y_{k+1} + \gamma_k y_k. \end{aligned} \tag{2}$$

If f is L-smooth then take

$$\gamma_k = \frac{1 - \lambda_k}{\lambda_{k+1}}$$

$$\lambda_k = \frac{1 + \sqrt{1 + 4\lambda_{k-1}^2}}{2}$$

Theorem .15. *Given an L-smooth function f, the sequence $\{f(x_k)\}$ produced by Nesterov's method* (2) *satisfies*

$$f(y_k) - f(x^*) \leq \frac{2L\|x_0 - x^*\|^2}{k^2}$$

If f is L-Smooth and μ-strongly convex then set

$$\gamma_k = \frac{\sqrt{Q} - 1}{\sqrt{Q} + 1}, \text{ where } Q = \frac{L}{\mu}$$

Theorem .16. *Given a function f which is L-Smooth, μ-strongly convex, the sequence $\{f(x_k)\}$ produced by Nesterov's method* (2) *satisfies*

$$f(y_k) - f(x^*) \leq \frac{(\mu + L)}{2}\|x_0 - x^*\|^2 exp(\frac{-k}{Q})$$

7 Proximal Point Algorithm

Consider an optimization problem

$$\min_x f(x)$$

Where f is convex but non-smooth, *i.e* we cannot apply gradient descent. In such cases we can use the proximal point algorithm-

$$x_{k+1} = \text{prox}_{\gamma f}(x_k) = \underset{u}{\text{argmin}}\ \gamma f(u) + \frac{1}{2}\|x_k - u\|^2 \quad (3)$$

Since f is convex, $f(u) + \frac{1}{2}\|x_k - u\|^2$ is strongly convex, the global mimizer u^* is unique. We take this u^* as x_{k+1} and repeat the process. Doing this for multiple iterations will lead to a point that mimizes f.

Theorem .17. *Given a convex function f, the sequence x_k produced by the Proximal Point Algorithm 3 satisfies*

$$f(x_k) - f(x^*) \leq \frac{1}{2k}\|x_0 - x^*\|^2$$

8 Non Convex Optimization

8.1 Polyak-Lojasiewicz inequality

A differentiable function $f : \mathbb{R}^n \to \mathbb{R}$ satisfies Polyak-Lojasiewicz (P-L) inequality if there exist a scalar α such that

$$\frac{1}{2}\|\nabla f(x)\|^2 \geq \alpha(f(x) - f(x^*))$$

Where x^* is the global minmizer. α is called the scaling constant.

8.1.1 P-L implies all stationary points are global minimizers

Since x^* is the global minimizer-

$$\frac{1}{2}\|\nabla f(x)\|^2 \geq \alpha(f(x) - f(x^*)) \geq 0$$

Then, at any point x where $\nabla f(x) = 0$, we have

$$0 = \frac{1}{2}\|\nabla f(x)\|^2 \geq \alpha(f(x) - f(x^*)) \geq 0$$

Thus we have $f(x) = f(x^*)$, *i.e* x is a global minimizer.

8.2 Accelerated methods

We consider the optimization problem [3]

$$\underset{x \in \mathbb{R}^d}{\text{minimize}} f(x)$$

where $f : R^d \to R$ has L_1-Lipschitz continuous gradient and L_2-Lipschitz continuous Hessian, but may be non-convex. Stationary points are defined as the points x with sufficiently small gradient:

$$\|\nabla f(x)\| \leq \varepsilon$$

[3] proposes the use of two competing techniques for making progress on computing a stationary point.

Lemma .18. *Let f be $\sigma_1 > 0$-strongly convex and L_1-smooth. Let $\varepsilon > 0$ and let z_j denote the jth iterate of AGD(f, z_1, ε, L_1, σ_1) 11 If*

$$j \geq 1 + \sqrt{\frac{L_1}{\sigma_1}} \log\left(\frac{4L_1^2}{\sigma_1 \varepsilon^2}\right) \quad \textit{then} \quad \|\nabla f(z_j)\| \leq \varepsilon.$$

Lemma .19. *Let f be be $\min\{\sigma_1, 0\}$-almost convex and L_1-smooth. Let $\gamma \geq \sigma_1$ and let $0 < \gamma \leq L_1$. Then ALMOST-CONVEX-AGDf,z_1,ε,γ,L_1 12 returns a vector z such that $\|\nabla f(z)\| \leq \varepsilon$ and*

$$f(z_1) - f(z) \geq \min\left\{\gamma\|z - z_1\|^2, \frac{\varepsilon}{\sqrt{10}}\|z - z_1\|\right\}$$

Algorithm 11 Accelerated-Gradient-Descent

```
function AGD($f, y_1, \varepsilon, L_1, \sigma_1$)
    Set $\kappa = \frac{L_1}{\sigma_1}, z_1 = y_1$
    for $j = 1, 2, \ldots$ do
        if $\|\nabla f(x)\| \leq \varepsilon$ then
            return $y_j$
        end if
        $y_{j+1} = z_j - \frac{1}{L_1}\nabla f(z_j)$
        $z_{j+1} = (1 + \frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1})y_{j+1} - \frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}y_j$
    end for
end function
```

in time

$$O\left(\left(\sqrt{\frac{L_1}{\gamma}} + \frac{\sqrt{\gamma L_1}}{\varepsilon^2}(f(z_1) - f(z))\right)\log\left(2 + \frac{L_1^3}{\gamma^2\varepsilon^2}\right)\right).$$

Algorithm 12 Almost Convex Accelerated-Gradient-Descent

```
function ALMOST-CONVEX-AGD($f, z_1, \varepsilon, \gamma, L_1$)
    for $j = 1, 2, \ldots$ do
        if $\|\nabla f(x)\| \leq \varepsilon$ then
            return $y_j$
        end if
        Let $g_j(z) = f(z) + \gamma\|z - z_j\|^2$
        $\varepsilon' = \varepsilon\sqrt{\gamma/(50(L_1 + 2\gamma))}$
        $z_{j+1} \leftarrow AGD(g_j, z_j, \varepsilon', l_1, \gamma)$
    end for
end function
```

Lemma .20. *Let the function f be L_1-smooth and have L_2-Lipschitz continuous Hessian, $\alpha > 0$, $0 < \delta < 1$ and $z_1 \in \mathcal{R}^d$. If we call NEGATIVE-CURVATURE-DESCENT(z_1, f, L_2, α, , δ) 13 then the algorithm terminates at iteration j for some*

$$j \leq 1 + \frac{12L_2^2(f(z_1) - f(z_j))}{\alpha^3} \leq 1 + \frac{12L_2^2}{\alpha^3},$$

and with probability at least $1 - \delta$

$$\lambda_{\min}(\nabla^2 f(z_j)) \geq -\alpha.$$

Furthermore, each iteration requires time at most

$$O\left(\left[1 + \sqrt{\frac{L_1}{\alpha}}\log\left(\frac{d}{\delta}\left(1 + 12\frac{L_2^2}{\alpha^3}\right)\right)\right]\right).$$

Algorithm 13 Negative-Curvature-Descent

```
function NEGATIVE-CURVATURE-DESCENT($z_1, f, L_2, \alpha, \Delta_f, \delta$)
    Set $\delta' = \delta/(1 + 12L_2^2\Delta_f/\alpha^3)$
    for $j = 1, 2, \dots$ do
        Find vector $v_j$ such that $\|v_j\| = 1$ and, with probability at least $1 - \delta'$,
            $$\lambda_{min}(\nabla^2 f(z_j)) \geq v_j^T \nabla^2 f(z_j) v_j - \alpha/2$$
        using a leading eigenvector computation
        if $v_j^T \nabla^2 f(z_j) v_j \leq \alpha/2$ then
            $z_{j+1} \leftarrow z_j - \frac{2|v_j^T \nabla^2 f(z_j) v_j|}{L_2} sign(v_j^T \nabla f(z_j)) v_j$
        else
            $z_j$
        end if
    end for
end function
```

Now, the ACCELERATED-NON-CONVEX-METHOD 14 uses 12 and 13 inorder to accelerate Smooth Non Convex optimization.

Algorithm 14 Acceleration of smooth non-linear optimization

```
function ACCELERATED-NON-CONVEX-METHOD($x_1, f, \varepsilon, L_1, L{-}2, \alpha, \Delta_f, \delta$)
    Set $K = 1 + \Delta_f(12L_2^2/\alpha^3 + \sqrt{10}L_2/(\alpha\varepsilon))$ and $\delta'' = \frac{\delta}{K}$
    for $j = 1, 2, \dots$ do
        if $\alpha < L_1$ then
            $\hat{x}_k \leftarrow$ NEGATIVE-CURVATURE-DESCENT($x_k, f, L_2, \alpha, \Delta_f, \delta''$)
        else
            $\hat{x}_k \leftarrow x_k$
        end if
        if $\|\nabla f(\hat{x}_k)\| \leq \varepsilon$ then
            return $\hat{x}_k$
        end if
        Set $f_k(x_k) = f(x) + L_1([\|x - \hat{x}_k\| - \alpha/L_2]_+)^2$
        $x_{k+1} \leftarrow$ ALMOST-CONVEX-AGD($f_k, \hat{x}_k, \varepsilon/2, 3\alpha, 5L_1$)
    end for
end function
```

References

[1] Elad Hazan. Introduction to online convex optimization. *CoRR*, abs/1909.05207, 2019.

[2] Euhanna Ghadimi, Hamid Reza Feyzmahdavian, and Mikael Johansson. Global convergence of the heavy-ball method for convex optimization, 2014.

[3] Yair Carmon, John C. Duchi, Oliver Hinder, and Aaron Sidford. Accelerated methods for nonconvex optimization. *SIAM Journal on Optimization*, 28(2):1751–1772, 2018.