Speeding Up Gradient Descent - Kaitlyn Chen, Dhruv Gautam, Samhith Kakarla

1a) In "A Method of Solving a Convex Programming Problem with Convergence Rate $O(1/k^2)$," Nesterov proposed a new method for solving convex optimization problems that converged faster than previously established methods. The main idea of Nesterovs accelerated gradient descent is to look ahead before you make the next step. The main assumption Nestrov makes on the function f to optimize is that f is a differentiable, convex, smooth function that's L-smooth and μ-strongly convex. The first function Nesterov shows to optimize via accelerated gradient descent is a convex function $f(x)$ that is L-smooth and μ-strongly convex such that the optimization problem is $\min\{f(x) \mid x \in E\}$ with a nonempty set X^* of minima. Then he considers the extremal problem $\min\{F(\vec{f}(x)) \mid x \in Q\}$ where Q is a convex closed set in E, and $F(u)$ where $u \in \mathbb{R}^m$ is a convex func on $\mathbb{R}^m$, positive homogenous of degree one, and $\vec{f}(x)$ is a vector of convex continuously differentiable functions on E. Nesterov's technique is to take the weighted average of the current iterate and a momentum iterate from the previous iteration. This weighted average is used as the next iterate. The update rule at the k^{th} step is $y_{k+1} = x_k + (a_k - 1)(x_k - x_{k-1})/a_{k+1}$ where $a_k = \frac{1}{2} a_{k+1}$, $x_k = y_k - a_k \nabla f(y_k)$, and $a_{k+1} = (1 + \sqrt{4a_k^2 + 1})/2$. Then at T iterations, the upper-bound on $f(\vec{x}_T) - \min_{\vec{x} \in \mathbb{R}^n} f(\vec{x})$ is $4L\|y_0 - x^*\|_2^2/(k+2)^2$ therefore $f(\vec{x}_T) - \min_{\vec{x} \in \mathbb{R}^n} f(\vec{x}) \leq 4L\|y_0 - x^*\|_2^2/(k+2)^2$ for any $k \geq 0$. Nesterovs accelerated gradient descent leads to a convergence rate of $O(1/k^2)$ for strongly convex functions as opposed to the slower $O(1/k)$ convergence rate achieved by standard gradient descent. Additionally Nesterov's method still has a faster convergence rate even for problems where the condition number of the Hessian can be large. Nesterov's method is also robust to noise therefore effective to many applications.

1b) In "Linear Coupling: An ultimate Unification of Gradient and Mirror Descent" Zhu and Orecchia proposed a new method to reconstruct Nestrov's accelerated gradient descent called linear coupling. This idea combines gradient descent (yeilding primal progress) and mirror descent (yeilding dual progress to obtain faster first order methods. This idea works since gradient and mirror descent are complementary. This is such because gradient descent is primal since you decrease the objective function iteratively as much as possible (while decrease stays guaranteed) till you converge. This approach ignores the dual problem, which mirror descent rather focuses on. Mirror descent develops lower bounds on the optimum point and they get tighter and tighter; these are global bounds unlike gradient descent's local leaps. Their method works for any unconstrained or constrained norm. The main assumptions made in AGM on the function f to optimize is that f is a differentiable and convex function on Q that is L-Smooth with respect to $\|\cdot\|$. For Strong convexity of AGM, they assume $f(\cdot)$ is both σ- strongly convex and L-Smooth with respect to $\|\cdot\|_2$. These functions are shown to be optimized via AGM. After T iterations of AGM, the upper bound on $f(\vec{y}_T) - \min_{\vec{x} \in Q} f(\vec{x})$ is $\frac{4\theta L}{(T+1)^2}$ where θ is any upper bound on $V_{x_0}(x^*)$. This paper shows how linear coupling has advantages over both gradient and mirror descent. For instance it can can handle non-euclidean geometries which gradient descent can't, it can handle non-smooth optimization problems which mirror descent can't, and it achieves convergence at rate of $O(1/k^2)$ just like Nesterov's accelerated gradient descent. This paper as you can see is connected with Nesterov's as it proves a similar version of accelerated gradient descent but, Zhu and Orecchia prove a more broad case application of accelerated gradient descent. Their method of AGM is also extended to non-convex problems.

2a) $D_h(\vec{y}, \vec{x}) = h(\vec{y}) - h(\vec{x}) - \langle \nabla h(\vec{x}), \vec{y} - \vec{x} \rangle$

$\nabla_{\vec{y}} D_h(\vec{y}, \vec{x}) = \nabla_{\vec{y}} h(\vec{y}) - \nabla_{\vec{x}} h(\vec{x})$

$\nabla^2_{\vec{y}} D_h(\vec{y}, \vec{x}) = \nabla^2_{\vec{y}} h(\vec{y})$

$\Rightarrow \nabla^2_{\vec{y}} D_h(\vec{y}, \vec{x}) - \alpha I \succeq 0$

$h(\vec{y})$ is given a α-strongly convex, thus $D_h(\vec{y}; \vec{x})$ is α-strongly convex

2b) $D_h(\vec{u}; \vec{x}) - D_h(\vec{u}; \vec{y}) - D_h(\vec{y}; \vec{x})$

$= h(\vec{u}) - h(\vec{x}) - \langle \nabla h(\vec{x}), \vec{u} - \vec{x} \rangle - h(\vec{u}) + h(\vec{y}) + \langle \nabla h(\vec{y}), \vec{u} - \vec{y} \rangle - h(\vec{y}) + h(\vec{x}) + \langle \nabla h(\vec{x}), \vec{y} - \vec{x} \rangle$

$= -\nabla h(\vec{x})^T \vec{u} + \nabla h(\vec{x})^T \vec{x} + \nabla h(\vec{y})^T \vec{u} - \nabla h(\vec{y})^T \vec{y} + \nabla h(\vec{x})^T \vec{y} - \nabla h(\vec{x})^T \vec{x}$

$= \langle \nabla h(\vec{x}) - \nabla h(\vec{y}), \vec{y} - \vec{u} \rangle$

2c) $h(\vec{x}) = \sum_{i=1}^{n} (x_i \log(x_i) - x_i)$

$\nabla h(\vec{x}) = \log(\vec{x})$

$\vec{z}_{k+1} = \text{Mirr}(\eta \vec{g}_k; \vec{z}) = \underset{\vec{z}_{k+1} \in X}{\text{argmin}} \langle \eta \vec{g}_k, \vec{z}_{k+1} \rangle + D_h(\vec{z}_{k+1}, \vec{z}_k)$

St. $\sum(\vec{z}_{k+1})_i = 1$

$-(\vec{z}_{k+1})_i \le 0 \quad \forall i = 1, \dots n$

$= \underset{\vec{z}_{k+1} \in X}{\text{argmin}} L(\vec{z}_{k+1}, \vec{\nu})$

$= \underset{\vec{z}_{k+1} \in X}{\text{argmin}} \langle \eta \vec{g}_k, \vec{z}_{k+1} \rangle + D_h(\vec{z}_{k+1}, \vec{z}_k) + \nu(\sum(z_{k+1})_i - 1) + \vec{\lambda}^T \vec{z}_{k+1}$

$= \underset{\vec{z}_{k+1} \in X}{\text{argmin}} \eta \vec{g}_k^T \vec{z}_{k+1} + h(\vec{z}_{k+1}) - h(\vec{z}_k) - \langle \nabla h(\vec{z}_k), \vec{z}_{k+1} - \vec{z}_k \rangle + \nu(\sum(z_{k+1})_i - 1) + \vec{\lambda}^T \vec{z}_{k+1}$

$\nabla_{\vec{z}_{k+1}} (\eta \vec{g}_k^T \vec{z}_{k+1} + h(\vec{z}_{k+1}) - \langle \nabla h(\vec{z}_k), \vec{z}_{k+1} \rangle + \nu(\sum(z_{k+1})_i) + \vec{\lambda}^T \vec{z}_{k+1}) = 0$

$\eta \vec{g}_k + \nabla_{\vec{z}_{k+1}} h(\vec{z}_{k+1}) - \nabla h(\vec{z}_k) + \nu \vec{1} + \vec{\lambda} = 0$

$\eta \vec{g}_k + \log(\vec{z}_{k+1}) - \log(\vec{z}_k) + \nu \vec{1} + \vec{\lambda} = 0$

$\log(\vec{z}_{k+1}) = \log(\vec{z}_k) - \nu \vec{1} - \eta \vec{g}_k - \vec{\lambda}$

$\vec{z}_{k+1}^* = e^{\log(\vec{z}_k)} e^{-\nu \vec{1}} e^{-\eta \vec{g}_k} e^{-\vec{\lambda}}$

$\vec{z}_{k+1} = \vec{z}_k e^{-\eta \vec{g}_k} e^{-\nu \vec{1}} e^{-\vec{\lambda}}$

$(\vec{z}_{k+1})_i = (\vec{z}_k)_i e^{-\eta (\vec{g}_k)_i} e^{-\nu} e^{-\lambda_i}$

$= (\vec{z}_k)_i e^{-\eta (\vec{g}_k)_i} e^{-\nu}$

$(\vec{z}_{k+1})_i^* = \dfrac{(z_k)_i e^{-\eta (\vec{g}_k)_i}}{\sum_j (\vec{z}_k)_j e^{-\eta (g_k)_i}}$

↳ if $(\vec{z}_k)_i$ doesn't violate the $(\vec{z}_{k+1})_i \ge 0$ constraint then $(\vec{z}_k)_i e^{-\eta (\vec{g}_k)_i} e^{-\nu} e^{-\lambda_i} \ge 0 \Rightarrow$ can eliminate $\vec{\lambda}$ Lagrangian multiplier

↳ $\sum_i (\vec{z}_{k+1})_i = 1$

$\sum_i (\vec{z}_k)_i e^{-\nu} e^{-\eta (\vec{g}_k)_i} = 1$

$e^{-\nu} = \dfrac{1}{\sum_i (\vec{z}_k)_i e^{-\eta (g_k)_i}}$

2d) $h(\vec{x}) = \frac{1}{2} \|\vec{x}\|_2^2$

$\nabla h(\vec{x}) = \vec{x}$

$\text{Mirr}(\eta \vec{g}; \vec{x}) \to \underset{\vec{z} \in \mathbb{R}^n}{\text{argmin}} \eta \langle \vec{g}, \vec{z} \rangle + D_h(\vec{z}, \vec{x})$

$\underset{\vec{z} \in \mathbb{R}^n}{\text{argmin}} \eta \langle \nabla f(\vec{x}), \vec{z} \rangle + h(\vec{z}) - h(\vec{x}) - \langle \nabla h(\vec{x}), \vec{z} - \vec{x} \rangle$

$\nabla_{\vec{z}} [\eta \langle \nabla f(\vec{x}), \vec{z} \rangle + h(\vec{z}) - \nabla h(\vec{x})^T \vec{z}] = 0$

$\eta \nabla f(\vec{x}) + \nabla h(\vec{z}) - \nabla h(\vec{x}) = 0$

$\eta \vec{g} + \vec{z} - \vec{x} = 0$

$\vec{z} = \vec{x} - \eta \vec{g}$

$D_h(\vec{y}, \vec{x}) = h(\vec{y}) - h(\vec{x}) - \langle \nabla h(\vec{x}), \vec{y} - \vec{x} \rangle$

$= \frac{1}{2} \|\vec{y}\|_2^2 - \frac{1}{2} \|\vec{x}\|_2^2 - \langle \vec{x}, \vec{y} - \vec{x} \rangle$

$$= \tfrac{1}{2}\|\vec{y}\|_2^2 - \tfrac{1}{2}\|\vec{x}\|_2^2 - \vec{x}^T\vec{y} + \|\vec{x}\|_2^2$$
$$= \tfrac{1}{2}\|\vec{y}\|_2^2 + \tfrac{1}{2}\|\vec{x}\|_2^2 - \vec{x}^T\vec{y}$$
$$= \tfrac{1}{2}\|\vec{y} - \vec{x}\|_2^2$$

2e) $\text{Reg}_k(\vec{u}) = \langle \eta_k \vec{g}_k, \vec{x}_k - \vec{u} \rangle$

$= \langle \eta_k \vec{g}_k, \vec{x}_k - \vec{x}_{k+1} \rangle + \langle \eta_k \vec{g}_k, \vec{x}_{k+1} - \vec{u} \rangle$

$= \langle \eta_k \vec{g}_k, \vec{x}_k - \vec{x}_{k+1} \rangle + \langle \eta_k \nabla f(\vec{x}_k), \vec{x}_{k+1} - \vec{u} \rangle$

$= \langle \eta_k \vec{g}_k, \vec{x}_k - \vec{x}_{k+1} \rangle + \langle -\nabla D_h(\vec{x}_{k+1}, \vec{x}_k), \vec{x}_{k+1} - \vec{u} \rangle$

$= \langle \eta_k \vec{g}_k, \vec{x}_k - \vec{x}_{k+1} \rangle + \langle \nabla(-h(\vec{x}_{k+1}) + h(\vec{x}_k) + \langle \nabla h(\vec{x}_k), \vec{x}_{k+1} - \vec{x}_k \rangle), \vec{x}_{k+1} - \vec{u} \rangle$

$= \langle \eta_k \vec{g}_k, \vec{x}_k - \vec{x}_{k+1} \rangle + \langle -\nabla h(\vec{x}_{k+1}) + \nabla h(\vec{x}_k) + \cancel{\nabla h(\vec{x}_k)^T \vec{x}_{k+1} - \nabla h(\vec{x}_k)^T \vec{x}_k}, \vec{x}_{k+1} - \vec{u} \rangle$

$= \langle \eta_k \vec{g}_k, \vec{x}_k - \vec{x}_{k+1} \rangle + D_h(\vec{u}; \vec{x}_k) - D_h(\vec{u}; \vec{x}_{k+1}) - D_h(\vec{x}_{k+1}; \vec{x}_k)$ ↘ By Bregman Divergence

$= \langle \eta_k \vec{g}_k, \vec{x}_k - \vec{x}_{k+1} \rangle + D_h(\vec{u}; \vec{x}_k) - D_h(\vec{u}; \vec{x}_{k+1}) - \tfrac{1}{2}\|\vec{x}_k - \vec{x}_{k+1}\|_2^2$

$= \langle \eta_k \vec{g}_k, \vec{x}_k - \vec{x}_{k+1} \rangle - \tfrac{1}{2}\|\vec{x}_k - \vec{x}_{k+1}\|_2^2 + D_h(\vec{u}; \vec{x}_k) - D_h(\vec{u}; \vec{x}_{k+1})$

$\leq \|\eta_k \vec{g}_k\|_2 \|\vec{x}_k - \vec{x}_{k+1}\|_2 - \tfrac{1}{2}\|\vec{x}_k - \vec{x}_{k+1}\|_2^2 + D_h(\vec{u}; \vec{x}_k) - D_h(\vec{u}; \vec{x}_{k+1})$ ↘ By cauchy-Shwarz Inequality

$\leq \frac{\|\eta_k \vec{g}_k\|_2^2 + \|\vec{x}_k - \vec{x}_{k+1}\|_2^2}{2} - \frac{\|\vec{x}_k - \vec{x}_{k+1}\|_2^2}{2} + D_h(\vec{u}; \vec{x}_k) - D_h(\vec{u}; \vec{x}_{k+1})$ ↘ By AM-GM → $\sqrt{xy} \leq \frac{x+y}{2}$

$\Rightarrow \sqrt{x^2 y^2} \leq \frac{x^2+y^2}{2}$

$= \frac{\|\eta_k \vec{g}_k\|_2^2}{2} + D_h(\vec{u}; \vec{x}_k) - D_h(\vec{u}; \vec{x}_{k+1})$

2f) $\text{TotalReg}_T(\vec{u}) = \sum_{k=0}^{T-1} \langle \eta_k \vec{g}_k, \vec{x}_k - \vec{u} \rangle \leq \sum_{k=0}^{T-1} \frac{\eta_k^2 \|\vec{g}_k\|_2^2}{2} + D_h(\vec{u}, \vec{x}_k) - D_h(\vec{u}, \vec{x}_{k+1})$

Telescoping from $k=0, k=1, \ldots k=T-1$ →

$\cancel{\tfrac{1}{2}\|\vec{u}\|_2^2} + \cancel{\tfrac{1}{2}\|\vec{x}_0\|_2^2} - \vec{u}^T\vec{x}_0 \; \cancel{-\tfrac{1}{2}\|\vec{x}_1\|_2^2} + \cancel{\vec{u}^T\vec{x}_1}$

$+ \cancel{\tfrac{1}{2}\|\vec{x}_1\|_2^2} + \cancel{\tfrac{1}{2}\|\vec{u}\|_2^2} - \cancel{\vec{u}^T\vec{x}_1} \; \cancel{-\tfrac{1}{2}\|\vec{x}_2\|_2^2} + \cancel{\vec{u}^T\vec{x}_2} + \ldots$

$\cancel{\tfrac{1}{2}\|\vec{x}_{T-1}\|_2^2} + \cancel{\tfrac{1}{2}\|\vec{u}\|_2^2} - \cancel{\vec{u}^T\vec{x}_{T-1}} \; \cancel{-\tfrac{1}{2}\|\vec{u}\|_2^2} + \tfrac{1}{2}\|\vec{x}_T\|_2^2 + \vec{u}^T\vec{x}_T$

$\leq \sum_{k=0}^{T-1} \frac{\eta_k^2 \|g_k\|_2^2}{2} + \left(D_h(\vec{u}, \vec{x}_0) - D_h(\vec{u}, \vec{x}_T)\right)$

$\leq \sum_{k=0}^{T-1} \eta_k^2 \|g_k\|_2^2 + D_h(\vec{u}, \vec{x}_0)$ ↘ $D_h(\vec{u}, \vec{x}_T) \geq 0$

2g) $\bar{x}_T = \frac{1}{T}\sum_{i=0}^{T-1} \vec{x}_i$ Assuming f is h

By the convexity of $\frac{1}{2}\|\vec{x}\|_2^2 = h(\vec{x})$,

$h(\bar{x}) \leq \frac{1}{T}\sum_{i=0}^{T-1} h(\vec{x}_i)$

Thus, we can also subtract some point $\vec{v}$ and bound it with

$h(\bar{x}) - h(\vec{v}) \leq \frac{1}{T}\sum_{i=0}^{T-1} \langle \nabla h(\vec{x}_i), \vec{x}_i - \vec{v} \rangle$

$\eta^2 \times T(h(\bar{x}) - h(\vec{v})) \leq \text{Total Reg}_T(\vec{v})$

Because η is positive,

$T\eta(h(\bar{x}) - h(\vec{v})) \leq \text{Total Reg}(\vec{v})$

Assume $\vec{v}$ is x^*

$f(\bar{x}_T) \leq f(x^*) + \frac{1}{T\eta}\underbrace{\text{Total Reg}(\vec{v})}_{\text{By part (f)}}$

$\leq f(\vec{x}^*) + \frac{1}{T\eta}\left(\eta^2 \sum_{k=0}^{T-1} \|\vec{g}_k\|_2^2 + D_h(\vec{x}^*; \vec{x}_0)/\eta\right)$

□

2h) WTD: $f(\bar{x}_T) \leq f(\vec{x}^*) + \eta L^2 + \frac{1}{2\eta T}\|\vec{x}_0 - \vec{x}^*\|_2^2$ We know $\|\nabla f(\vec{x})\|_2 \leq L \quad \forall \vec{x} \in X$

$f(\vec{x}^*) + \frac{1}{T\eta}\sum_{k=0}^{T-1} \|\vec{g}_k\|_2^2 + D_h(\vec{x}^*; \vec{x}_0)/\eta$

Let $\frac{1}{2}\|\vec{x}_0 - \vec{x}^*\|_2^2$ be some upper bound on $D_n(\vec{x}^* \,|\, \vec{x}_0)$

& due to L-lipschitz $\|\nabla f(\vec{x})\|_2 \le L$

Thus, $f(\bar{x}_T) \le f(\vec{x}^*) + \eta L^2 + \frac{1}{2\eta T}\|\vec{x}_0 - \vec{x}^*\|_2^2$

$$\min_{\eta} f(\vec{x}^*) + \eta L^2 + \frac{1}{2\eta T}\|\vec{x}_0 - \vec{x}^*\|_2^2$$

$$\nabla_\eta\left(f(\vec{x}^*) + \eta L^2 + \frac{1}{2\eta T}\|\vec{x}_0 - \vec{x}^*\|_2^2\right) = 0$$

$$L^2 - \frac{\|\vec{x}_0 - \vec{x}^*\|_2^2}{2\eta^2 T} = 0$$

$$L^2 = \frac{\|\vec{x}_0 - \vec{x}^*\|_2^2}{2\eta^2 T}$$

$$2\eta^2 T L^2 = \|\vec{x}_0 - \vec{x}^*\|_2^2$$

$$\eta^2 = \frac{\|\vec{x}_0 - \vec{x}^*\|_2^2}{2TL^2}$$

$$\eta^* = \frac{\|\vec{x}_0 - \vec{x}^*\|_2}{\sqrt{2}\sqrt{T}L}$$

$$f(\bar{x}_T) \le f(\vec{x}^*) + \eta^* L^2 + \frac{1}{2\eta^* T}\|\vec{x}_0 - \vec{x}^*\|_2^2$$

$$\le f(\vec{x}^*) + \frac{\|\vec{x}_0 - \vec{x}^*\|_2 L^2}{\sqrt{2}\sqrt{T}L} + \frac{\|\vec{x}_0 - \vec{x}^*\|_2^2}{2T}\cdot\frac{\sqrt{2}\sqrt{T}L}{\|\vec{x}_0 - \vec{x}^*\|_2}$$

$$\le f(\vec{x}^*) + \frac{\|\vec{x}_0 - \vec{x}^*\|_2 L}{\sqrt{2}\sqrt{T}} + \frac{\|\vec{x}_0 - \vec{x}^*\|_2\sqrt{2}\sqrt{T}L}{2T}$$

$$\le f(\vec{x}^*) + \frac{\|\vec{x}_0 - \vec{x}^*\|_2 L}{\sqrt{2}\sqrt{T}}\left(\frac{\sqrt{2}\sqrt{T}}{\sqrt{2}\sqrt{T}}\right) + \frac{\|\vec{x}_0 - \vec{x}^*\|_2\sqrt{2}\sqrt{T}L}{2T}$$

$$\le f(\vec{x}^*) + \frac{\|\vec{x}_0 - \vec{x}^*\|_2 L\sqrt{2}\sqrt{T} + \|\vec{x}_0 - \vec{x}^*\|_2\sqrt{2}\sqrt{T}L}{2T}$$

$$\le f(\vec{x}^*) + \frac{2\|\vec{x}_0 - \vec{x}^*\|_2 L\sqrt{2}\sqrt{T}}{2T}$$

$$f(\bar{x}_T) \le f(\vec{x}^*) + \frac{\sqrt{2}L\|\vec{x}_0 - \vec{x}^*\|_2}{\sqrt{T}}$$

3a) $\langle \eta_{k+1}\nabla f(\vec{x}_{k+1}), \vec{z}_k - \vec{u}\rangle \le \frac{\eta_{k+1}^2\|\nabla f(\vec{x}_{k+1})\|_2^2}{2} + D(\vec{u}, z_k) - D(\vec{u}, \vec{z}_{k+1})$ ← follows from 2e since the mirror step $\vec{z}_{k+1} = \text{Mirr}(\eta_k\nabla f(\vec{x}_{k+1}), \vec{z}_k)$ is almost exactly the same at that from 2e, $\vec{x}_{k+1} = \text{Mirr}(\eta_k\nabla f(\vec{x}_k), \vec{x}_k)$ except different variable names. 2e would still work if $\vec{g}_k$ was somethin other than $\nabla f(\vec{z}_k)$

3b) $\langle \eta_{k+1}\nabla f(\vec{x}_{k+1}), \vec{x}_{k+1} - \vec{u}\rangle = -\eta_{k+1}\langle \nabla f(x_{k+1}), \vec{z}_k - \vec{x}_{k+1}\rangle + \eta_{k+1}\langle \nabla f(\vec{x}_{k+1}), \vec{z}_k - \vec{u}\rangle$

$$\le -\eta_{k+1}\langle \nabla f(x_{k+1}), \vec{z}_k - \vec{x}_{k+1}\rangle + \frac{\eta_{k+1}^2}{2}\|\nabla f(\vec{x}_{k+1})\|_2^2 + D(\vec{u}\,|\,\vec{z}_k) - D(\vec{u}\,|\,\vec{z}_{k+1})$$

Choose $\vec{x}_{k+1}$ that satisfies $\tau_k(\vec{x}_{k+1} - \vec{z}_k) = (1-\tau_k)(\vec{y}_k - \vec{x}_{k+1})$

$$\le \eta_{k+1}\langle \nabla f(x_{k+1}), \vec{x}_{k+1} - \vec{z}_k\rangle + \frac{\eta_{k+1}^2}{2}\|\nabla f(\vec{x}_{k+1})\|_2^2 + D(\vec{u}\,|\,\vec{z}_k) - D(\vec{u}\,|\,\vec{z}_{k+1})$$

$$\langle \eta_{k+1}\nabla f(\vec{x}_{k+1}), \vec{x}_{k+1} - \vec{u}\rangle = \frac{(1-\tau_k)\alpha_{k+1}}{\tau_k}\langle \nabla f(x_{k+1}), \vec{y}_k - \vec{x}_{k+1}\rangle + \frac{\eta_{k+1}^2}{2}\|\nabla f(\vec{x}_{k+1})\|_2^2 + D(\vec{u}\,|\,\vec{z}_k) - D(\vec{u}\,|\,\vec{z}_{k+1})$$

$$\le \frac{(1-\tau_k)\alpha_{k+1}}{\tau_k}(f(\vec{y}_k) - f(\vec{x}_{k+1})) + \frac{\eta_{k+1}^2}{2}\|\nabla f(\vec{x}_{k+1})\|_2^2 + D(\vec{u}\,|\,\vec{z}_k) - D(\vec{u}\,|\,\vec{z}_{k+1})$$

by f convexity & $1-\tau_k \ge 0$

$$\langle \eta_{k+1}\nabla f(\vec{x}_{k+1}), \vec{x}_{k+1} - \vec{u}\rangle \le \frac{(1-\tau_k)\alpha_{k+1}}{\tau_k}(f(\vec{y}_k) - f(\vec{x}_{k+1})) + \alpha_{k+1}^2 L(f(\vec{x}_{k+1}) - f(\vec{y}_{k+1})) + D(\vec{u}\,|\,\vec{z}_k) - D(\vec{u}\,|\,\vec{z}_{k+1})$$

3c) Lemma 4.3 & 4.2

In the previous part we found the inequality that

$$\frac{(1-\tau_k)\eta_{k+1}}{\tau_k}\langle \nabla f(x_{k+1}), y_k - x_{k+1}\rangle + \eta_{k+1}\langle \nabla f(x_{k+1}), z_k - u\rangle$$

$$\leq \frac{(1-\tau_u)\eta_{u+1}}{\tau_u}\left(f(y_u)-f(x_{u+1})\right)+\eta_{u+1}\langle \nabla f(x_{u+1}), z_u-\vec{v}\rangle$$

which by part (b) is also

$$\leq \frac{(1-\tau_u)\eta_{u+1}}{\tau_u}\left(f(y_u)-f(x_{u+1})\right)+\eta_{u+1}^2 L\left(f(x_{u+1})-f(y_{u+1})\right)+D(\vec{v};z_u)-D(\vec{v};z_{u+1})$$

IF $\tau_u = \frac{1}{\eta_{u+1}L}$,

$$= \left(\eta_{u+1}^2 L-\eta_{u+1}\right)f(y_u)-\left(\eta_{u+1}^2 L\right)f(y_{u+1})+\eta_{u+1}f(x_{u+1})+D(\vec{v};z_u)-D(\vec{v};z_{u+1})$$

We started off with

$$\langle \eta_{u+1}\nabla f(\vec{x}_{u+1}), \vec{z}_u-\vec{v}\rangle \geq \eta_{u+1}\left(f(x_{u+1})-f(\vec{v})\right)$$

and can thus upper-bound our statement with

$$\leq \eta_{u+1}F(\vec{v})$$

3d)

WTP: $f(\vec{y}_T) \leq f(\vec{x}^*)+\frac{2L\|\vec{x}^*-\vec{x}_0\|_2^2}{(T+1)^2}$

Because of the previous part, we can telescope if using where

$\eta_u^2 L = \eta_{u+1}^2 L-\eta_{u+1}$ & $\tau_u = \frac{1}{\eta_{u+1}L} \in (0,1]$ ← Motivation from paper

$\eta_u = \frac{u+1}{2L}$

$\eta_u^2 L = \eta_{u+1}^2 L-\eta_{u+1}+\frac{1}{4L}$

Sum over $u=0,\ldots,T-1$ to get

Using part (c) with summation →

$$\underbrace{\eta_T^2 L f(y_T)}_{0 \text{ since}}+\sum_{k=1}^{T-1}\frac{1}{4L}f(y_u)+\left(D_R(v;z_T)-D_f(v;z_0)\right)\leq\sum_{u=1}^{T}\eta_u f(\vec{v})$$

IF $\vec{v}=\vec{x}^*$, $\sum_{k=1}^{T-1}\eta_u = \frac{T(T+3)}{4L}$,

$D_f(\vec{v};\vec{z}_T)$ is positive / zero,

$D_f(\vec{v};\vec{z}_0)$ is bounded by $\frac{1}{2}\|\vec{x}_0-\vec{x}^*\|_2^2$,

& $f(\vec{y}_u) \geq f(\vec{x}^*)$

This leads into the inequality,

$$\frac{(T+1)^2}{4L^2} Lf(y_T) \leq \left(\frac{T(T+3)}{4L} - \frac{T-1}{4L}\right) f(x^*) + \frac{1}{2}\left\|\vec{x}_0 - \vec{x}^*\right\|_2^2$$

$$f(y_T) \leq \frac{4L}{(T+1)^2}\left(\frac{T^2+3T-T+1}{4L}\right) f(x^*) + \frac{1}{2}\left\|\vec{x}_0 - \vec{x}^*\right\|_2^2 \left(\frac{4L}{(T+1)^2}\right)$$

$$f(y_T) \leq f(x^*) + \frac{2L\left\|\vec{x}_0 - \vec{x}^*\right\|_2^2}{(T+1)^2}$$ □

3e) With a learning rate of 0.001 accerated gradient descent out-performed gradient descent, mirror descent, and adagrad, converging to a minimum in fewer iterations.

EECS 127 PROJECT: SPEEDING UP GRADIENT DESCENT EXTENSION - "A GEOMETRIC ALTERNATIVE TO NESTEROVS ACCELERATED GRADIENT DESCENT"

KAITLYN CHEN, DHRUV GAUTAM, AND SAMHITH KAKARLA

1. Introduction

The paper we chose to study was S. Bubeck, Y. T. Lee, and M. Singh's, "A geometric alternative to Nesterov's accelerated gradient descent". In the paper, they propose an alternative method to solve unconstrained optimization problems with a smooth and strongly convex objective function. They prove that the proposed optimal method, which is inspired by the ellipsoid method, achieves the same, or potentially better optimal rate of convergence as Nesterov's accelerated gradient descent for unconstrained optimization. This paper, printed in 2015, proved by example that there are still many open problems in the field of accelerated gradient descent. This includes its application to non-convex optimization problems and understanding under what conditions it can still converge to a minimum and Its application to more general settings such as constrained or distributed optimization. Practical applications of accelerated gradient descent are still open problems; noisy data can adversely affect performance and convergence in different cases greatly. Finally, an effective means of finding the optimal step size has also troubled researchers as it is not known in advance how the objective function would be shaped. The optimal step size could be influenced by many factors such as the curvature of the objective function itself and can be very different for different problems. Finding this optimal step size and determining on a broad scale what influences the optimality is definitely one of the largest open problems related to accelerated gradient descent.

2. Methodology

We began by reading and analyzing the research paper and understanding the mathematical intuition behind this optimal algorithm they proposed. On a base level, the algorithm assumes that the objective function to be optimized is unconstrained, smooth, and strongly convex. The algorithm that they call geometric descent works by shrinking the optimal search space at each iteration using roughly a combination of the ellipsoid method and gradient descent. The geometric intuition for this approach stems from the fact that if we have 2 intersecting balls, the intersecting space and radii shrink much faster if both balls shrink at the same absolute amount compared to if only one ball shrinks. Following this, the algorithm finds the next iterate by computing the minimum enclosing ball of the intersection of two balls centered at the current iterate and the previous iterate. This minimum enclosing ball narrows down the optimal search space at each step and provides the direction in which to take the next step. Then the algorithm chooses the next x by a line search which ensures that $f(x_k) \leq f(x_k - 1)$.

We implemented this algorithm starting with the linear search algorithm. Which was defined as such:

$$linesearch(x, y) = \operatorname*{argmin}_{t \in \mathbb{R}} f(x + t(y - x)).$$

We utilized the scipy optimize library in Python to implement the linear search function that given some function f and 2 vectors x and y could return the argmin of the given function as defined above. Then we implemented a method to find the Minimum Enclosing Ball of the Intersection to Two Balls. This input for this method would be a ball centered at x_A with radius r_A and a ball centered at x_B with radius r_B and would return the minimum enclosing ball that would cover the intersection of the two balls. Finally,

we implemented our geometric descent algorithm stemming from the following pseudo-code from the paper.

Algorithm 2: Geometric Descent Method (GeoD)

Input: parameters α and initial points x_0.
$x_0^+ = \texttt{line\_search}(x_0, x_0 - \nabla f(x_0))$.
$c_0 = x_0 - \alpha^{-1}\nabla f(x_0)$.
$R_0^2 = \frac{|\nabla f(x_0)|^2}{\alpha^2} - \frac{2}{\alpha}\left(f(x_0) - f(x_0^+)\right)$.
for $i \leftarrow 1, 2, \cdots$ **do**
  Combining Step:
  $x_k = \texttt{line\_search}(x_{k-1}^+, c_{k-1})$.
  Gradient Step:
  $x_k^+ = \texttt{line\_search}(x_k, x_k - \nabla f(x_k))$.
  Ellipsoid Step:
  $x_A = x_k - \alpha^{-1}\nabla f(x_k)$. $R_A^2 = \frac{|\nabla f(x_k)|^2}{\alpha^2} - \frac{2}{\alpha}\left(f(x_k) - f(x_k^+)\right)$.
  $x_B = c_{k-1}$. $R_B^2 = R_{k-1}^2 - \frac{2}{\alpha}\left(f(x_{k-1}^+) - f(x_k^+)\right)$.
  Let $B(c_k, R_k^2)$ is the minimum enclosing ball of $B(x_A, R_A^2) \cap B(x_B, R_B^2)$.
end
Output: x_T.

We amended their pseudo - code a little bit to fit our project format and to return all iterates rather than just the last step (important for actually getting the results out).

The paper mentions that like all other iterative algorithms that utilize only the gradient information, in the worst case, the algorithm cannot converge at a rate faster than $1 - \Theta(\beta - 1/2)$. Where β is the smoothness parameter. Still, it is worth noting that the geometric descent algorithm converged at this rate using far less memory than some of the other algorithms. This has implications in the real world when data sets get gigantic. Additionally, their method integrates zeroth and first order information in a unique manner, which obviously helps in real world applications. This is a large takeaway from these accelerated descent papers; although they reach similar descent times compared to Nesterov's, their superior implementations and easy to understand formats help in the true implementations and implications of their algorithms.

3. Results

Let us begin by examining the theorems that were used to prove the functionality of the algorithm. One important theorem was

$$For\ any\ k\ \geq 0,\ one\ has\ x^* \in B({c_k}^2, {R_k}^2),\ {R_{k+1}}^2 \leq (1 - \frac{1}{\sqrt{\kappa}}){R_k}^2,\ and\ thus:$$

$$|x^* - c_k|^2 \leq (1 - \frac{1}{\sqrt{\kappa}})^k\ {R_0}^2$$

Here they are essentially trying to prove that at each iteration the new step we take and the resulting ball originating at that point is smaller than the previous step. This is pivotal to proving the algorithm works, as it works by shrinking the optimal search space at each step. This is a core idea of the entire geometric descent method. They prove this useful theorem by proving a stronger theorem that the squared radius of the ball that encloses the intersection is smaller than:

$$(1 - \frac{1}{\sqrt{\kappa}}){R_k}^2 - \frac{2}{\alpha}(f({x_{k+1}}^+) - f(x^*))$$

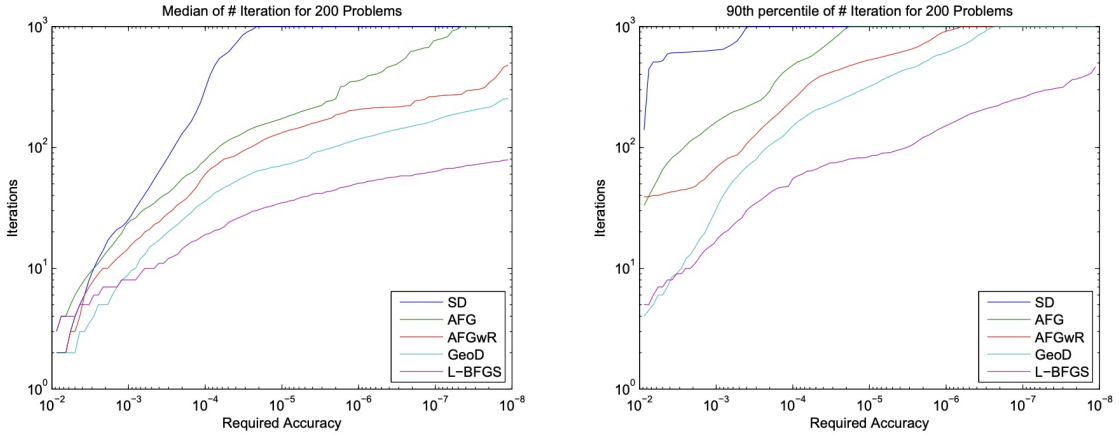
using induction.

Another theorem that was useful in proving the method was this lemma:

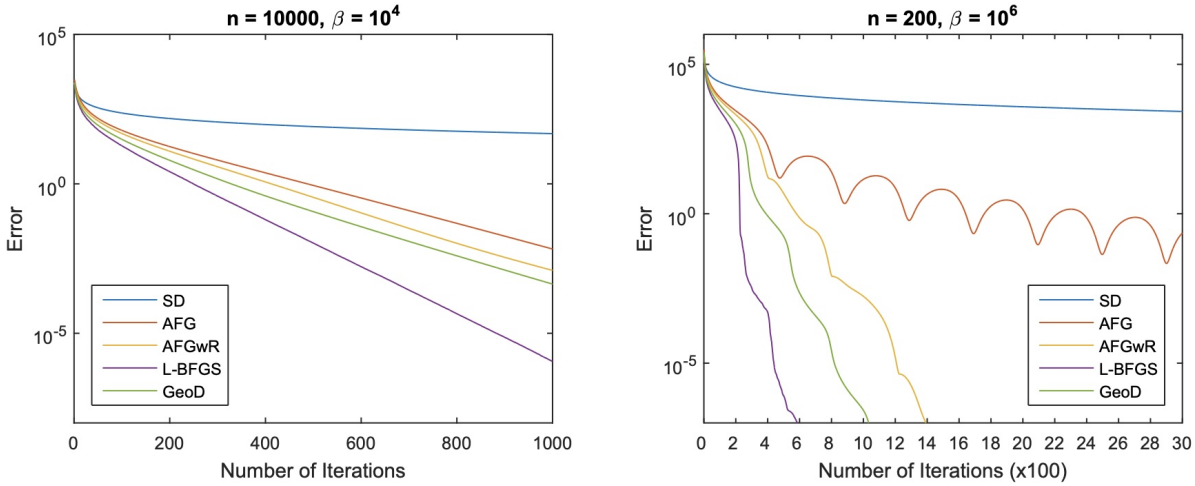
Let $a \in \mathbb{R}^{\aleph}$ *and* $\varepsilon \in (0,1), g \in \mathbb{R}_+$ *Assume that* $|a| \geq g$ *. Then there exists* $c \in \mathbb{R}^{\aleph}$ *such that for any* $\delta > 0$

$$B(0, 1 - \varepsilon g^2 - \delta) \cap B(a, g^2(1-\varepsilon) - \delta) \subset B(c, 1 - \sqrt{\varepsilon} - \delta)$$

Their resulting algorithm, as expected, seemed to outperform many other standard approaches to solving unconstrained convex optimization problems such as steepest descent (SD), accelerated full gradient method (AFG) and accelerated full gradient method with adaptive restart (AFGwR)

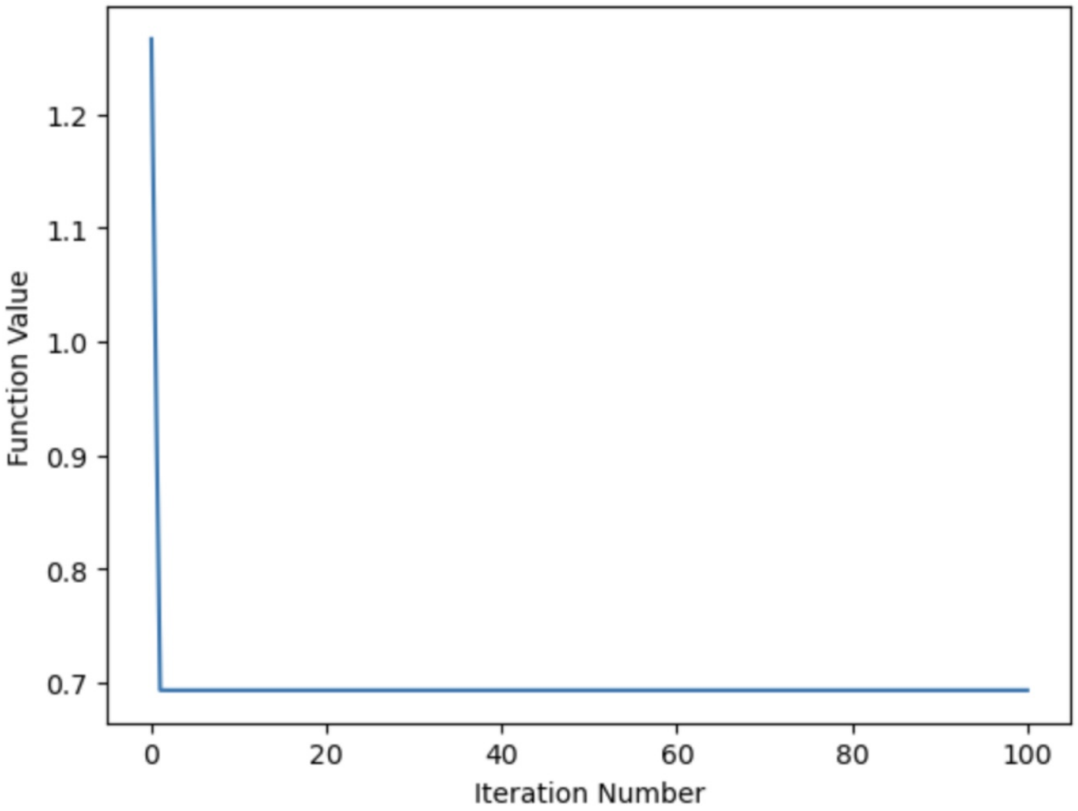


Here are the error guarantees of the different algorithms that were tested. As one can see, GeoD performs extremely well comparatively to the other algorithms. Specifically, with GeoD develops an miniscule error hundreds/thousands of iterations earlier than most of the other algorithms (that still have great performance in comparison to steepest descent). These graphs represent it:

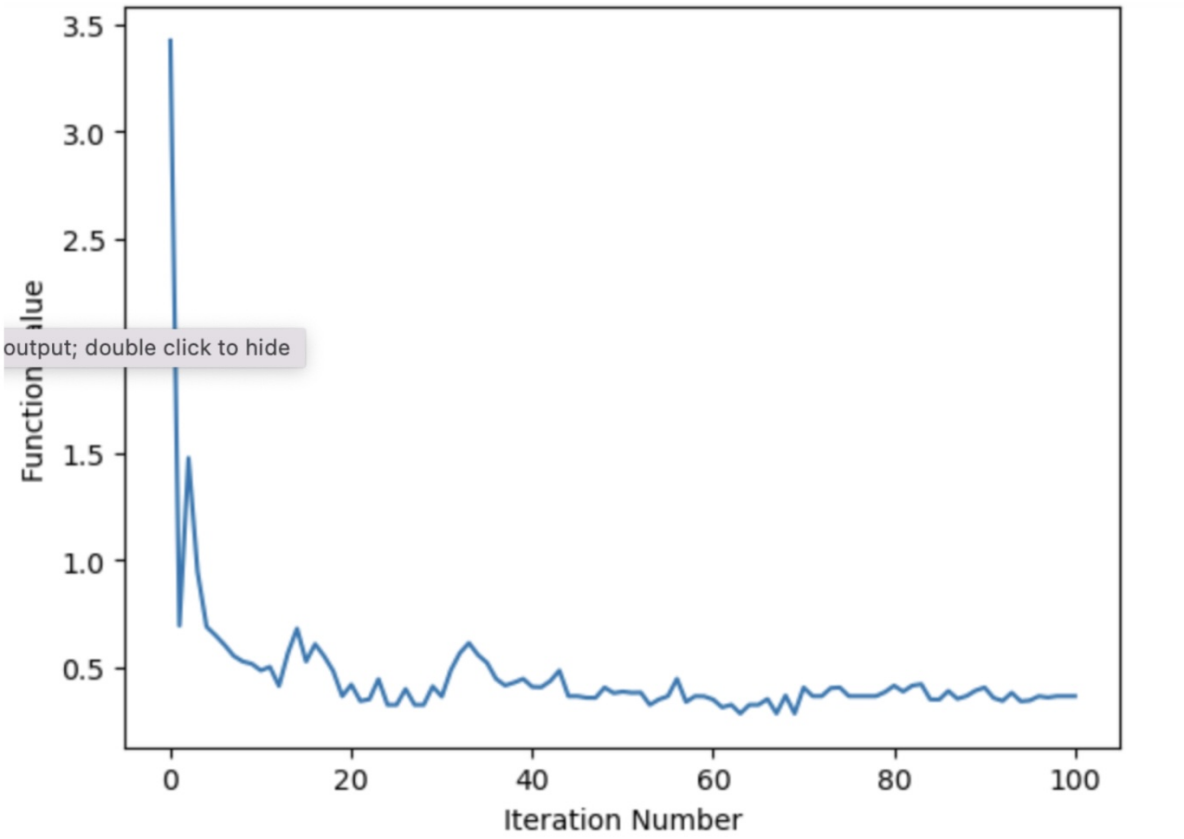


Our results were similar with Geometric descent outperforming gradient and mirror descent and performing approximately as well as Nesterov's accelerated descent which was the intended goal.

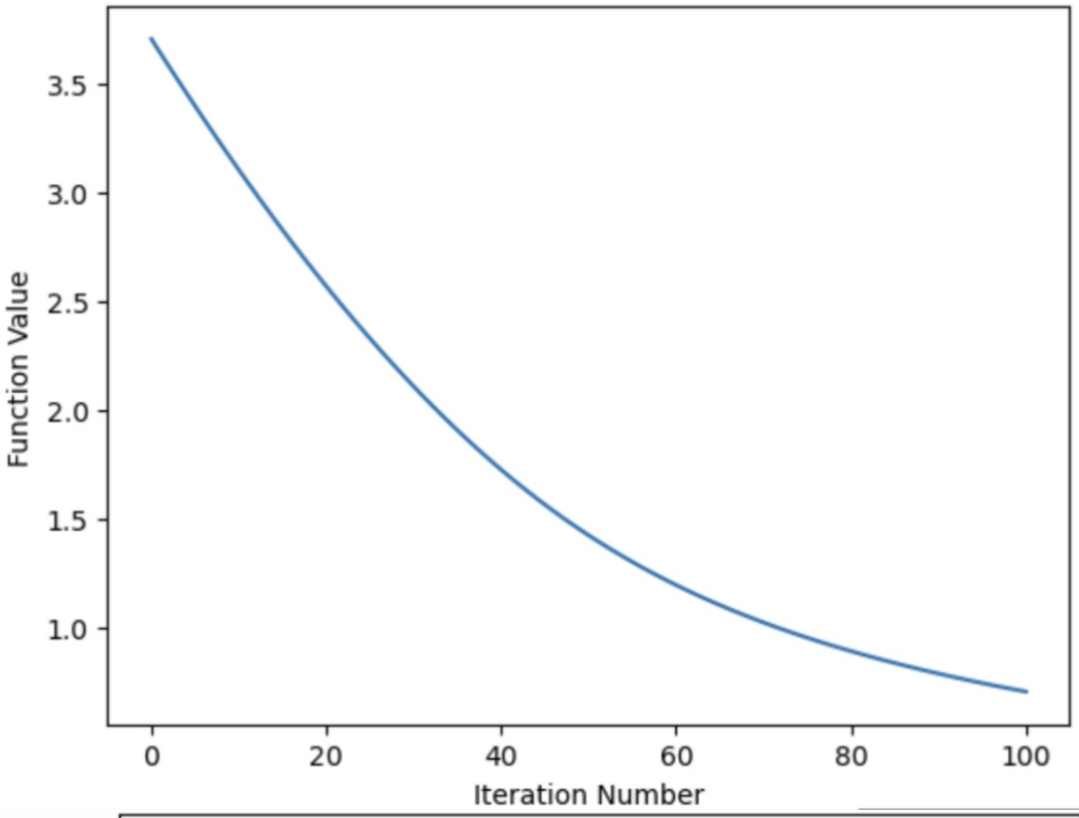
Graph of geometric descent (GeoD):

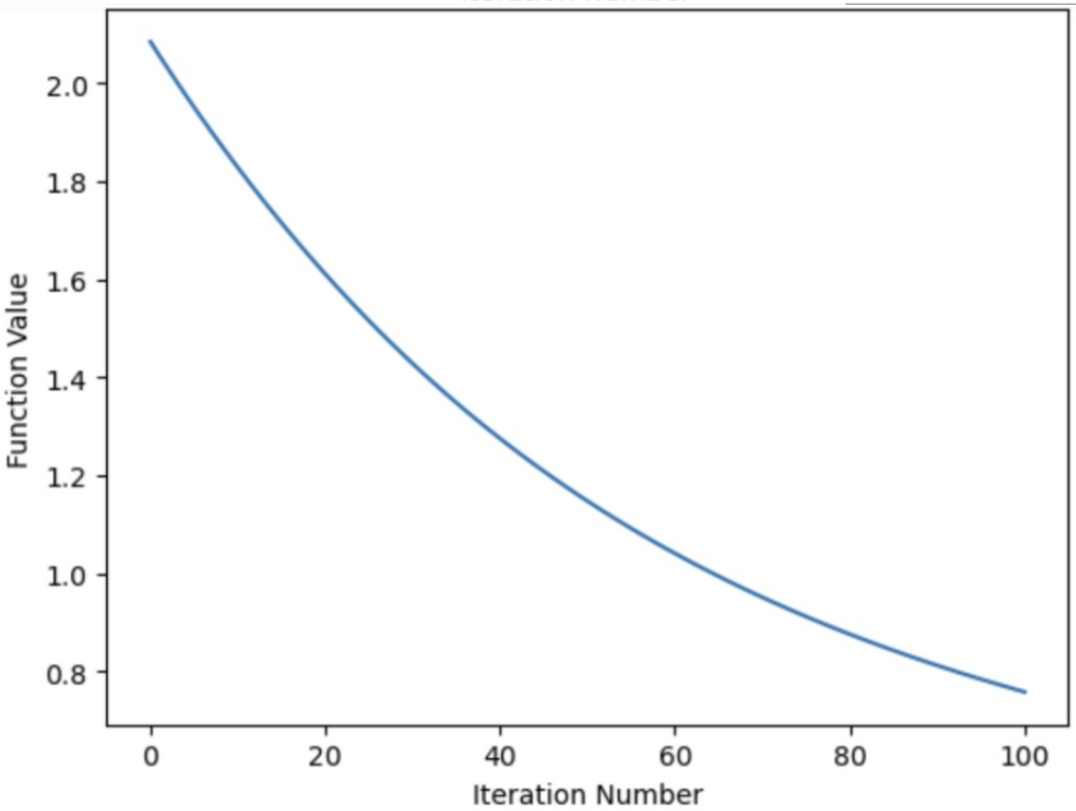


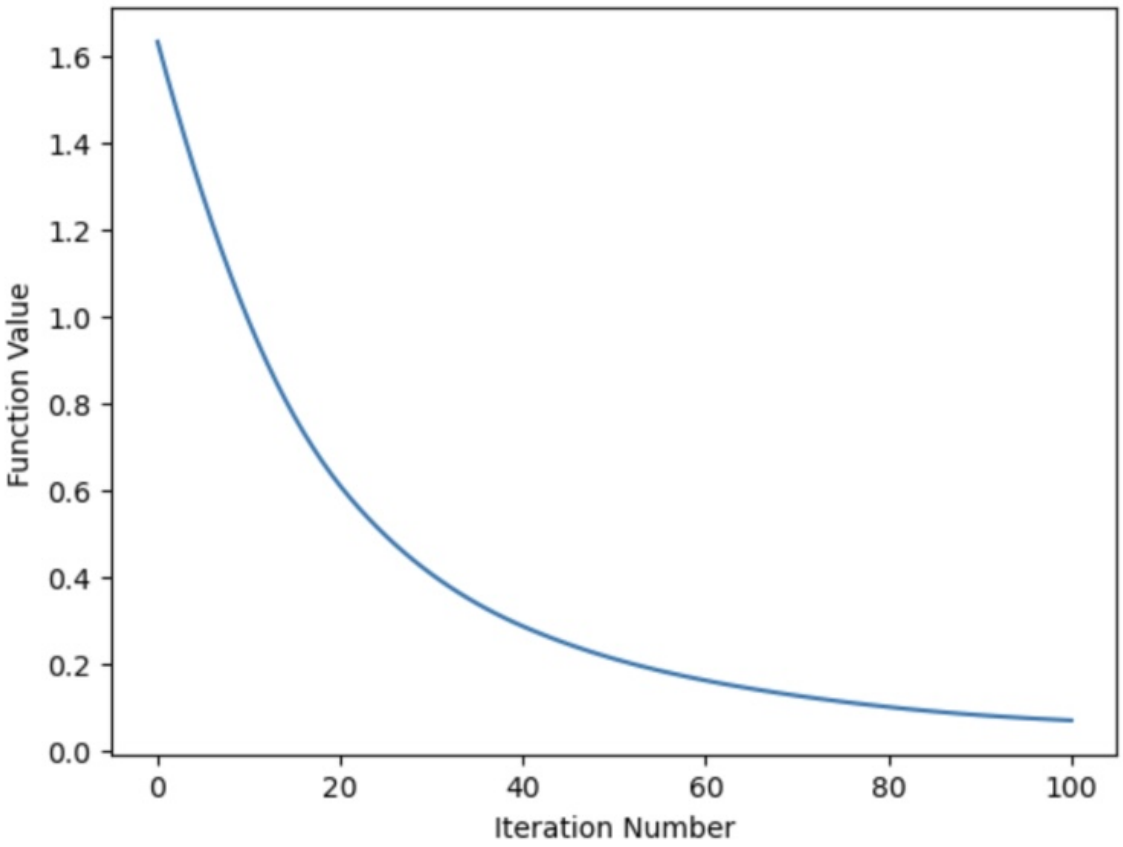
Graph of Nesterov's accelerated descent:



Graphs of mirror, gradient descent, and Adaptive gradient method (adagrad) respectively:


Function Value
3.5
3.0
2.5
2.0
1.5
1.0
0
20
40
60
80
100
Iteration Number


Function Value
2.0
1.8
1.6
1.4
1.2
1.0
0.8
0
20
40
60
80
100
Iteration Number



As we can see with the same step size/ learning rate of 0.001 accelerated descent and geometric descent outperform regular gradient/mirror descent and converge to the minimum much faster. They also outperformed the adagrad method.