Glocal Smoothness: Line Search can really help!

Curtis Fox CURTFOX@CS.UBC.CA
Mark Schmidt SCHMIDTM@CS.UBC.CA
University of British Columbia

Abstract

Iteration complexities are bounds on the number of iterations of an algorithm. Iteration complexities for first-order numerical optimization algorithms are typically stated in terms of a global Lipschitz constant of the gradient, and near-optimal results are achieved using fixed step sizes. But many objective functions that arise in practice have regions with small Lipschitz constants where larger step sizes can be used. Many local Lipschitz assumptions have thus been proposed, which lead to results showing that adaptive step sizes and/or line searches yield improved convergence rates over fixed step sizes. However, these faster rates tend to depend on the iterates of the algorithm, which makes it difficult to compare the iteration complexities of different methods. We consider a simple characterization of global and local smoothness that only depends on properties of the function. This allows upper bounds on iteration complexities in terms of problem-dependent constants, which allows us to compare iteration complexities between algorithms. Under this assumption it is straightforward to show the advantages of line searches over fixed step sizes, and that in some settings gradient descent with line search has a better iteration complexity than accelerated gradient methods with fixed step sizes.

1. Setting the Step Size: Theory vs. Practice

Machine learning models are typically trained using numerical optimization algorithms. The simplest algorithm used is gradient descent [5], which on iteration t takes steps of the form

$$w_{t+1} = w_t - \eta_t \nabla f(w_t), \tag{1}$$

for some positive step size η_t. The simplest way to set the step size η_t is to use a constant value throughout training. Other methods have also been proposed in order to give faster convergence in practice, including line searches [1] or adaptive step sizes such as the Polyak step size [25]. To prove theoretical guarantees on the convergence of gradient descent using different step sizes, the function being optimized is often assumed to have a globally Lipschitz continuous gradient (also known as being L-smooth).

Definition 1 *A function f has an L-Lipschitz continuous gradient if $\forall x, y \in \mathbb{R}^d$, then:*

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

Unfortunately, there is often a disconnect between the step sizes that work in theory and those that work well in practical applications. To illustrate the disconnect between theory practice, contrast using a fixed step size of $\eta_t = \frac{1}{L}$ with choosing the step using line optimization (LO) to minimize the function,

$$\eta_t \in \arg\min_{\eta} f(w_t - \eta \nabla f(w_t)). \tag{2}$$

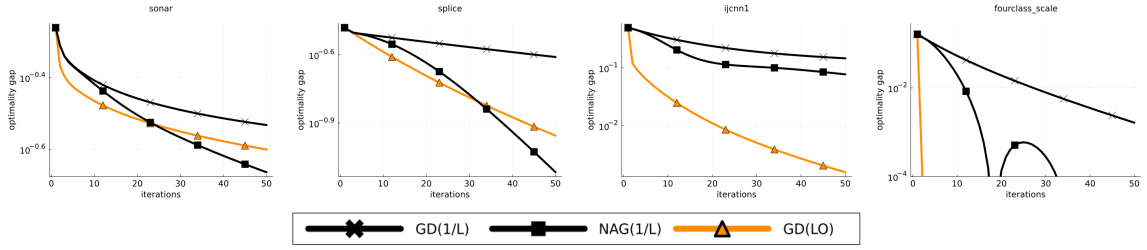

Figure 1: Logistic regression on 4 different datasets showing the optimality gap throughout training for gradient descent (GD) and Nesterov's accelerated gradient (NAG) method with a fixed step size of $\frac{1}{L}$, along with gradient descent with line optimization (LO). From left to right, we have a problem where GD(LO) has a similar rate to GD(1/L), a problem where GD(LO) is converging faster than GD(1/L), and two problems where GD(LO) is also converging faster than NAG(1/L).

We refer to gradient descent with $\eta_t = \frac{1}{L}$ as GD(1/L) and gradient descent with η_t set using (2) as GD(LO). Standard analyses of GD(1/L) and GD(LO) give nearly-identical theoretical convergence rates [11], but GD(LO) typically converges faster in practice. Further, GD(LO) has a worse theoretical convergence rate than Nesterov's accelerated gradient (NAG) method using a step size of $\frac{1}{L}$ [19]. But experimentally, GD(LO) converges faster than NAG(1/L) on many problems (see Figure 1).

One reason GD(LO) can converge faster than GD(1/L) and NAG(1/L) is that it can use much larger step sizes than $\frac{1}{L}$. These larger step sizes are possible because, in a region around each w_t, the Lipschitz smoothness assumption (1) may hold with a smaller value of the constant L. Many works have analyzed first-order methods under local measures of L (see Section 2). However, these analyses generally depend on the iterates w_t of the algorithm, making it difficult to compare convergence rates between algorithms; different algorithms will take differing paths during training so have different local L values. Thus, it is hard to use these assumptions to explain why GD(LO) can outperform NAG(1/L).

In Section 3, we introduce a "glocal" smoothness assumption that augments global smoothness with a measure of local smoothness near the solution. In particular, a function is (L, L^*, δ)-glocal smooth if it is globally L-smooth and locally L_*-smooth when the sub-optimality is at most δ. This characterization of global and local smoothness leads to iteration complexities that only depend on the properties of the function being optimized, and not on the precise path taken by the algorithm. Similar to previous local smoothness assumptions, we show under glocal smoothness that GD(LO) has an improved iteration complexity over GD(1/L). But under the glocal assumption, we also give conditions under which GD(LO) has a better iteration complexity than NAG(1/L). While using LO is not always practical, in Section 4 we consider practical step sizes under the glocal assumption. Finally, in Section 5 we discuss related research directions.

2. Related Work: how much does line search and local smoothness help?

It may seem possible that the performance of GD(LO) may be explained by a better analysis under global smoothness. Indeed, recent work by de Klerk et al. [7] gives tight rates for GD(LO) that are

faster than for GD(1/L). However, these rates do not explain why GD(LO) can converge much faster than NAG(1/L).

Many works define a notion of local smoothness based on the iterates w_t [3, 13–16, 22, 26, 30]. These local smoothness conditions depend on the lines between successive iterations w_{t-1} and w_t, balls around the w_t, the convex hull of the w_t, or sub-level sets of the $f(w_t)$. Under these assumptions we can show that GD(LO) converges faster than GD(1/L). The reason for GD(LO)'s faster convergence is that it can use larger step sizes that exploit the local smoothness, while the step sizes of GD(1/L) are based on the global smoothness constant. However, we cannot easily compare GD(LO) and NAG(1/L) using these measures. Different algorithms will have different iterates w_t, so the particular local smoothness constants cannot be compared between algorithms.

Many algorithms use estimates of local smoothness to speed convergence [8, 12, 14, 20, 23, 24, 27–29, 31]. We can use the assumptions of the previous paragraph to show that such methods can converge faster than methods that do not exploit local smoothness. However, existing tools do not allow us to compare between different algorithms exploiting local smoothness.

The most-closely related works to our glocal-smoothness assumption are works that assume local smoothness but do not assume global smoothness [13, 23, 24, 31]. These works tend to focus on achieving global convergence despite the lack of global smoothness. In contrast, our focus is on exploring assumptions under which it is easy to compare the iteration complexities of different algorithms.

Finally, we note that glocal-smoothness is a special case of the general framework of Curtis and Robinson [6]. They propose partitioning the search space of an algorithm into different regions that satisfy different assumptions, and analyzing the progress an algorithm makes in each region. Our work considers the simpler case where we consider just two regions: the whole space and a sub-level set. We believe that this is an important special case as it leads to simple analyses that better reflect practical performance. Further, we expect that algorithms that perform well under glocal smoothness should have good performance under other measures of local smoothness.

3. Glocal Smoothness

In order to more easily compare optimization algorithms that may exploit local smoothness, we propose a notion of global-local smoothness, which we call "glocal smoothness":

Definition 2 *A function f is glocally (L, L_*, δ)-smooth if f is globally L-smooth, and locally L_*-smooth for all $x \in \mathbb{R}^d$ such that $f(x) - f^* \leq \delta$ for some $\delta > 0$.*

Assumptions of this type have been explored for speeding convergence [29] and analyzing the quality of local optima [17]. Note that $L_* \leq L$ since L^* is measured over a subset of the space, but that for some problems we have $L_* << L$. For example, the standard bound for binary logistic regression with a data matrix X is $(1/4)\|X\|^2$ [4]. The 1/4 factor is the upper bound on $p(y_i = +1)p(y_i = -1)$ for labels y_i. However, if near solutions we have $p(y_i = +1) > 0.99$ or $p(y_i = -1) > .99$ for all i, then L_* is around $(1/100)\|X\|^2$. This allows GD(LO) to eventually take steps that are 25-times larger than those used by GD(1/L) yet still decrease the function.

3.1. Iteration Complexity under Glocal Smoothness

To illustrate how glocal smoothness allows comparisons between algorithms, we consider the case of strongly-convex functions:

Definition 3 *A function f is μ-strongly convex for some $\mu > 0$ if $\forall x, y \in \mathbb{R}^d$,*

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

Under strong convexity and global smoothness assumptions, GD(1/L) reaches an accuracy of ϵ after $(L/\mu)\log((f(w_0) - f^*)/\epsilon)$ iterations. This is worse than the $O(\sqrt{L/\mu}\log((f(w_0) - f^*)/\epsilon))$ iteration complexity of NAG(1/L) with appropriate algorithm parameter settings [see 18]. Under strong convexity and glocal smoothness, we have the following iteration complexity for GD(LO):

Theorem 4 *Assume that f is glocally (L, L_*, δ)-smooth and μ-strongly convex. For all $t \geq 0$, let w_t be the iterates of gradient descent as defined in (1) with step-size η_t given by (2). Then for $\delta > \epsilon$,*

$$f(w_T) - f^* \leq \epsilon \quad \textit{for all} \quad T \geq \frac{L}{\mu}\log\left(\frac{f(w_0) - f^*}{\delta}\right) + \frac{L_*}{\mu}\log\left(\frac{\delta}{\epsilon}\right).$$

The proof is given in Appendix B.1. The rate of GD(LO) under glocal smoothness is faster than the rate of GD(1/L) whenever we have $L_* < L$. Indeed, the analysis proceeds by arguing that GD(LO) makes at least as much progress as GD(1/L) globally and makes at least as much progress as GD(1/L^*) locally (without needing to know any of L, L^*, or δ).

It is notable that rate of GD(LO) under glocal smoothness is faster than NAG(1/L) if

$$\frac{L_*}{L} \leq \frac{(\sqrt{\kappa})^{-1}\log(1/\epsilon) - \log(1/\delta)}{\log(1/\epsilon) - \log(1/\delta)}.$$

This is possible when $L_* < L$, $\delta > \epsilon$, and the condition number $\kappa = \frac{L}{\mu}$ is not too large. In words, we expect line search to outperform acceleration if the problem is not too badly conditioned and there is a large region around the minimizers with a smaller local smoothness constant than the global smoothness constant.

4. Practical Algorithms

Unfortunately, neither GD(1/L) or GD(LO) are practical algorithms in general as it may not be possible in practice to compute the Lipschitz constant L or perform LO. Under global smoothness, we can achieve the rate of GD(1/L) using a backtracking line search procedure [2]. This method starts with an initial guess L_0 of L, and doubles the value whenever a sufficient decrease condition is not satisfied. If $L_0 \leq L$, it achieves the GD(1/L) rate of $O((L/\mu)\log((f(w_0) - f^*)/\epsilon))$. Unfortunately, this backtracking method does not achieve a faster rate under glocal smoothness since it never decreases its guess of L (and does not increase the step size in the local region).

It is possible to obtain an improved iteration complexity over GD(1/L) in practice under glocal smoothness using a backtracking procedure with resets [26]. When the guess of L is reset to L_0 on each iteration, the rate of gradient descent is improved to $O((L/\mu)\log((f(w_0) - f^*)/\delta) + (\max\{L_*, L_0\}/\mu)\log(\delta/\epsilon))$ under the glocal framework. This backtracking-with-resets method achieves the fast rate of GD(LO) if $L_0 \leq L_*$. However, this method may require significant backtracking on each iteration while it has a worse complexity than GD(LO) if $L_0 > L^*$.

We can achieve the rate of GD(LO) under glocal smoothness using a procedure that includes both a forwardtracking and a backtracking procedure [8]. Unfortunately, the iterations of such methods can be expensive since the step size could be increased or decreased many times within

each iteration of the algorithm. Fortunately, many practical heuristics exist to reduce the cost of this type of algorithm [see 21].

Finally, we note that it is possible to achieve a rate similar to GD(LO) under glocal smoothness without increasing the iteration cost if we know f^* and use the Polyak step size [25],

$$\eta_t = \frac{f(w_t) - f^*}{\|\nabla f(w_t)\|^2}, \tag{3}$$

Theorem 5 *Assume that f is glocally (L, L_*, δ)-smooth and μ-strongly convex. For all $t \geq 0$, let w_t be the iterates of gradient descent as defined in (1) with step-size η_t given by (3). Then for $\delta > \epsilon$,*

$$f(w_T) - f^* \leq \epsilon \quad \textit{for all} \quad T \geq 4\left(\frac{L}{\mu}\log\left(\frac{L}{2}\frac{\|w_0 - w_*\|^2}{\delta}\right) + \frac{L_*}{\mu}\log\left(\frac{\delta}{\epsilon}\right)\right).$$

See Appendix B.2 for the proof of this result. This result is similar to GD(LO), with a worse dependence inside the first logarithmic factor since the non-monotonicity of the Polyak step size means we may take more iterations to guarantee that we stay within the δ region.

5. Discussion

We focus on glocal smoothness and global strong convexity, but other global-local assumptions could be explored. For example, we could obtain faster rates if we allowed a larger local μ_* within the δ region. We could also relax strong-convexity to consider functions satisfying the Polyak-Łojasiewicz condition [see 11]. An alternate relaxation is to consider functions that are strongly-convex in the local region but only convex globally. For smooth convex functions the complexity of GD(1/L) is $O(L/\epsilon)$ and for NAG(1/L) it is $O(\sqrt{L/\epsilon})$. But under glocal smoothness and local strong convexity we obtain $O(L/\delta + (L_*/\mu*)\log(\delta/\epsilon))$ for GD(LO), which may be much smaller. However, it is less clear how to exploit a condition like glocal smoothness for non-convex functions.

Our analysis focuses on variants of GD, but glocal smoothness can be used to analyze other algorithms. For example, it can be inserted into existing analyses of coordinate descent and proximal-gradient methods to obtain better rates for those methods under LO. It would be interesting to explore accelerated and stochastic methods that adapt to glocal smoothness. A challenge with analyzing some stochastic methods is that they may leave the local δ region infinitely often.

Compared to popular global smoothness assumptions, we believe that performance under glocal smoothness better reflects the performance of numerical optimization algorithms in practice. Thus, we encourage theoreticians to adopt assumptions like glocal smoothness in order for their theoretical results to better reflect empirical performance. Further, a key advantage of glocal smoothness compared to many previous local smoothness assumptions is that glocal smoothness allows comparisons between algorithms. Indeed, we have given a precise condition under which "line search can really help" in the sense that using line search with the basic GD method leads to a faster convergence rate than using acceleration.

References

[1] Larry Armijo. Minimization of functions having lipschitz continuous first partial derivatives. *Pacific Journal of mathematics*, 16(1):1–3, 1966.

[2] Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences*, 2(1):183–202, 2009.

[3] Albert S. Berahas, Lindon Roberts, and Fred Roosta. Non-uniform smoothness for gradient descent. *CoRR*, abs/2311.08615, 2023.

[4] Dankmar Böhning. Multinomial logistic regression algorithm. *Ann. Inst. Stat. Math.*, 44(1): 197–200, 1992.

[5] Augustin Cauchy et al. Méthode générale pour la résolution des systemes d'équations simultanées. *Comp. Rend. Sci. Paris*, 25(1847):536–538, 1847.

[6] Frank E. Curtis and Daniel P. Robinson. Regional complexity analysis of algorithms for nonconvex smooth optimization. *Math. Program.*, 187(1):579–615, 2021.

[7] Etienne de Klerk, François Glineur, and Adrien B. Taylor. On the worst-case complexity of the gradient method with exact line search for smooth strongly convex functions. *Optim. Lett.*, 11(7):1185–1199, 2017.

[8] Sara Fridovich-Keil and Benjamin Recht. Choosing the step size: intuitive line search algorithms with efficient convergence. In *11th annual workshop on optimization for machine learning*, pages 1–21, 2019.

[9] Guillaume Garrigos and Robert M Gower. Handbook of convergence theorems for (stochastic) gradient methods. *arXiv preprint arXiv:2301.11235*, 2023.

[10] Elad Hazan and Sham Kakade. Revisiting the polyak step size. *arXiv preprint arXiv:1905.00313*, 2019.

[11] Hamed Karimi, Julie Nutini, and Mark Schmidt. Linear convergence of gradient and proximal-gradient methods under the polyak-łojasiewicz condition. In *Machine Learning and Knowledge Discovery in Databases - European Conference*, volume 9851 of *Lecture Notes in Computer Science*, pages 795–811. Springer, 2016.

[12] Yan Liu, Congying Han, and Tiande Guo. A class of stochastic variance reduced methods with an adaptive stepsize. *URL http://www. optimization-online. org/DB_FILE/2019/04/7170. pdf*, 2019.

[13] Zhaosong Lu and Sanyou Mei. Accelerated first-order methods for convex optimization with locally lipschitz continuous gradient. *SIAM J. Optim.*, 33(3):2275–2310, 2023.

[14] Yura Malitsky and Konstantin Mishchenko. Adaptive gradient descent without descent. In *International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 6702–6712. PMLR, 2020.

[15] Jincheng Mei, Yue Gao, Bo Dai, Csaba Szepesvári, and Dale Schuurmans. Leveraging non-uniformity in first-order non-convex optimization. In *International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 7555–7564. PMLR, 2021.

[16] Aaron Mishkin, Ahmed Khaled, Yuanhao Wang, Aaron Defazio, and Robert M. Gower. Directional smoothness and gradient methods: Convergence and adaptivity. *CoRR*, abs/2403.04081, 2024.

[17] Amirkeivan Mohtashami, Martin Jaggi, and Sebastian U Stich. Special properties of gradient descent with large learning rates. In *International Conference on Machine Learning*, pages 25082–25104. PMLR, 2023.

[18] Yurii Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2013.

[19] Yurii Nesterov et al. *Lectures on convex optimization*, volume 137. Springer, 2018.

[20] Yurii E. Nesterov. Gradient methods for minimizing composite functions. *Math. Program.*, 140(1):125–161, 2013.

[21] Jorge Nocedal and Stephen J Wright. *Numerical optimization*. Springer, 1999.

[22] Francesco Orabona. Normalized gradients for all. *CoRR*, abs/2308.05621, 2023.

[23] Jea-Hyun Park, Abner J. Salgado, and Steven M. Wise. Preconditioned accelerated gradient descent methods for locally lipschitz smooth objectives with applications to the solution of nonlinear pdes. *J. Sci. Comput.*, 89(1):17, 2021.

[24] Vivak Patel and Albert S Berahas. Gradient descent in the absence of global lipschitz continuity of the gradients. *arXiv preprint arXiv:2210.02418*, 2022.

[25] Boris T Polyak. Introduction to optimization, 1987.

[26] Katya Scheinberg, Donald Goldfarb, and Xi Bai. Fast first-order methods for composite convex optimization with backtracking. *Found. Comput. Math.*, 14(3):389–417, 2014.

[27] Mark Schmidt, Nicolas Le Roux, and Francis R. Bach. Minimizing finite sums with the stochastic average gradient. *Math. Program.*, 162(1-2):83–112, 2017.

[28] Zheng Shi, Abdurakhmon Sadiev, Nicolas Loizou, Peter Richtárik, and Martin Takác. AI-SARAH: adaptive and implicit stochastic recursive gradient methods. *Trans. Mach. Learn. Res.*, 2023, 2023.

[29] Daniel Vainsencher, Han Liu, and Tong Zhang. Local smoothness in variance reduced optimization. In *Advances in Neural Information Processing Systems*, pages 2179–2187, 2015.

[30] Hui Zhang and Wotao Yin. Gradient methods for convex minimization: better rates under weaker conditions. *CoRR*, abs/1303.4645, 2013.

[31] Junyu Zhang and Mingyi Hong. First-order algorithms without lipschitz gradient: A sequential local optimization approach. *INFORMS Journal on Optimization*, 2024.

Appendix A. Lemmas

Lemma 6 *Assume f is L-smooth. Then the following holds for the gradient descent algorithm:*

$$f(w_{t+1}) \leq f(w_t) - (\eta_t - \frac{L\eta_t^2}{2})\|\nabla f(w_t)\|^2$$

Proof See the proof of Lemma 2.28 in the work by Garrigos and Gower [9]. ■

Appendix B. Convergence Proofs

B.1. Glocally Smooth, Strongly Convex Convergence Proof

Proof of Theorem 4.

Proof This proof is based on the proof of Theorem 1 in Karimi et al. [11]. Assume $\epsilon < \delta$. Since the algorithm will never increase the function value on any iteration (which we show in our proof), once the iterates of the algorithm enter the region where f is locally L_*-smooth, the iterates will not leave this region again. Thus we can break up the proof into two parts. The first part shows the number of iterations of gradient descent required to reach δ accuracy of the optimal solution w^*, and thus enter the region where f is locally L_* smooth. The second part shows how many iterations of gradient descent are required to reach ϵ accuracy of the optimal solution.

Part 1: Since f is glocally $\{L, L_* \delta\}$-smooth, then f is globally L-smooth, and we can apply Lemma 6:

$$f(w_t) \leq f(w_{t-1}) - (\eta_{t-1} - \frac{L\eta_{t-1}^2}{2})\|\nabla f(w_{t-1})\|^2$$

Now using an exact line search guarantees at least as much progress on the objective function f as using a step size of $\frac{1}{L}$:

$$f(w_t) \leq f(w_{t-1}) - \frac{1}{2L}\|\nabla f(w_{t-1})\|^2$$

This implies that the function value decreases on each iteration. Since f is globally μ-strongly convex, then the PL-inequality holds [11], therefore:

$$f(w_t) \leq f(w_{t-1}) - \frac{\mu}{L}(f(w_{t-1}) - f^*)$$

Subtracting f^* from both sides:

$$f(w_t) - f^* \leq (1 - \frac{\mu}{L})(f(w_{t-1}) - f^*)$$

Applying the contraction recursively from $t = 0, \ldots, t_\delta$:

$$\begin{aligned} f(w_{t_\delta}) - f^* &\leq (1 - \frac{\mu}{L})^{t_\delta}(f(w_0) - f^*) \\ &\leq \delta \end{aligned}$$

since we want to compute the number of iterations required to reach an accuracy of δ. This corresponds to an iteration complexity of $t_\delta \geq (\frac{L}{\mu}\log(\frac{f(w_0)-f^*}{\delta}))$.

Part 2: This proof will look very similar to Part 1. Using the fact that f is glocally $\{L, L_*, \delta\}$-smooth, then f is locally L_* smooth, and we can apply Lemma 6:

$$f(w_t) \leq f(w_{t-1}) - (\eta_{t-1} - \frac{L_*\eta_{t-1}^2}{2})\|\nabla f(w_{t-1})\|^2$$

This holds when $f(w_t) - f^* \leq \delta$. Now using an exact line search guarantees at least as much progress on the objective function f as using a step size of $\frac{1}{L_*}$:

$$f(w_t) \leq f(w_{t-1}) - \frac{1}{2L_*}\|\nabla f(w_{t-1})\|^2$$

This implies that the function value decreases on each iteration. Since f is globally μ-strongly convex, then the PL-inequality holds [11], therefore:

$$f(w_t) \leq f(w_{t-1}) - \frac{\mu}{L_*}(f(w_{t-1}) - f^*)$$

Subtracting f^* from both sides:

$$f(w_t) - f^* \leq (1 - \frac{\mu}{L_*})(f(w_{t-1}) - f^*)$$

Note that the initial iterate in this phase is w_{t_δ}. Let $T = t_\delta + t_\epsilon$. Applying the contraction recursively from $t = t_\delta, \ldots, T$:

$$f(w_T) - f^* \leq (1 - \frac{\mu}{L_*})^{t_\epsilon}(f(w_{t_\delta}) - f^*)$$

Now from Part 1 we know $f(w_{t_\delta}) - f^* \leq \delta$, therefore:

$$\begin{aligned} f(w_T) - f^* &\leq (1 - \frac{\mu}{L_*})^{t_\epsilon}\delta \\ &\leq \epsilon \end{aligned}$$

since we want to compute the number of iterations required to reach an accuracy of ϵ. This corresponds to an iteration complexity of $t_\epsilon \geq (\frac{L_*}{\mu}\log(\frac{\delta}{\epsilon}))$.

Combining both parts, the overall iteration complexity is $T \geq (\frac{L}{\mu}\log(\frac{f(w_0)-f^*}{\delta}) + \frac{L_*}{\mu}\log(\frac{\delta}{\epsilon}))$. ■

B.2. Polyak Convergence Proof

Proof of Theorem 5.

Proof This proof is similar to a proof in Hazan and Kakade [10]. Assume $\epsilon < \delta$. We start by noting that if f is L-smooth, then:

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

Now letting $x = w_*$, where w_* is a minimizer of f, we end up with:

$$f(w) - f(w_*) \leq \frac{L}{2}\|w - w_*\|^2 \tag{4}$$

Therefore if we can show that $\frac{L}{2}\|w - w_*\|^2 \leq \delta$, this implies that $f(w) - f^* \leq \delta$. Using this result, and the fact that the algorithm will never increase the iterate distance on any iteration (which we show below), we split up the proof into two parts. The first part computes the number of iterations to reach δ accuracy of the optimal solution w^*, and the second computes the number of iterations to reach ϵ accuracy of the optimal solution.

Part 1: We start by rewriting $\|w_t - w_*\|^2$ as follows, using the gradient descent update:

$$\begin{aligned}\|w_t - w_*\|^2 &= \|(w_{t-1} - w_*) - \eta_{t-1}\nabla f(w_{t-1})\|^2 \\ &= \|w_{t-1} - w_*\|^2 - 2\eta_{t-1}\langle \nabla f(w_{t-1}), w_{t-1} - w_*\rangle + \eta_{t-1}^2\|\nabla f(w_{t-1})\|^2\end{aligned}$$

Since f is strongly convex, it is also convex, therefore:

$$\|w_t - w_*\|^2 \leq \|w_{t-1} - w_*\|^2 - 2\eta_{t-1}(f(w_{t-1}) - f^*) + \eta_{t-1}^2\|\nabla f(w_{t-1})\|^2$$

Substituting the Polyak step size update (3) for η_{t-1}:

$$\begin{aligned}\|w_t - w_*\|^2 &\leq \|w_{t-1} - w_*\|^2 - 2(\frac{f(w_{t-1}) - f^*}{\|\nabla f(w_{t-1})\|^2})(f(w_{t-1}) - f^*) + (\frac{f(w_{t-1}) - f^*}{\|\nabla f(w_{t-1})\|^2})^2\|\nabla f(w_{t-1})\|^2 \\ &= \|w_{t-1} - w_*\|^2 - \frac{(f(w_{t-1}) - f^*)^2}{\|\nabla f(w_{t-1})\|^2}\end{aligned}$$

Thus we guarantee a decrease in the iterate distance on each iteration of the algorithm. Now since f is μ-strongly convex and glocally $\{L, L_* \delta\}$-smooth (and therefore globally L-smooth):

$$\begin{aligned}\|w_t - w_*\|^2 &\leq \|w_{t-1} - w_*\|^2 - \frac{f(w_{t-1}) - f^*}{2L} \\ &\leq \|w_{t-1} - w_*\|^2 - \frac{\mu}{4L}\|w_{t-1} - w_*\|^2 \\ &= (1 - \frac{\mu}{4L})\|w_{t-1} - w_*\|^2\end{aligned}$$

Applying the contraction recursively from $t = 0, \ldots, t_\delta$:

$$\|w_{t_\delta} - w_*\|^2 \leq (1 - \frac{\mu}{4L})^{t_\delta}\|w_0 - w_*\|^2$$

Now multiplying both sides by $\frac{L}{2}$, we get that:

$$\frac{L}{2}\|w_{t_\delta} - w_*\|^2 \leq \frac{L}{2}(1 - \frac{\mu}{4L})^{t_\delta}\|w_0 - w_*\|^2$$

Now by (4), if the following holds:

$$\frac{L}{2}\|w_{t_\delta} - w_*\|^2 \leq \frac{L}{2}(1 - \frac{\mu}{4L})^{t_\delta}\|w_0 - w_*\|^2 \leq \delta \tag{5}$$

Then this implies that:

$$f(w_{t_\delta}) - f^* \leq \delta$$

as required. This corresponds to an iteration complexity of $t_\delta \geq 4(\frac{L}{\mu}\log(\frac{L}{2}\frac{\|w_0 - w_*\|^2}{\delta}))$.

Part 2: This proof will look very similar to Part 1. We start by rewriting $\|w_t - w_*\|^2$ as follows, using the gradient descent update:

$$\begin{aligned}\|w_t - w_*\|^2 &= \|(w_{t-1} - w_*) - \eta_{t-1}\nabla f(w_{t-1})\|^2 \\ &= \|w_{t-1} - w_*\|^2 - 2\eta_{t-1}\langle \nabla f(w_{t-1}), w_{t-1} - w_*\rangle + \eta_{t-1}^2\|\nabla f(w_{t-1})\|^2\end{aligned}$$

Since f is strongly convex, it is also convex, therefore:

$$\|w_t - w_*\|^2 \leq \|w_{t-1} - w_*\|^2 - 2\eta_{t-1}(f(w_{t-1}) - f^*) + \eta_{t-1}^2\|\nabla f(w_{t-1})\|^2$$

Substituting the Polyak step size update (3) for η_{t-1}:

$$\begin{aligned}\|w_t - w_*\|^2 &\leq \|w_{t-1} - w_*\|^2 - 2(\frac{f(w_{t-1}) - f^*}{\|\nabla f(w_{t-1})\|^2})(f(w_{t-1}) - f^*) + (\frac{f(w_{t-1}) - f^*}{\|\nabla f(w_{t-1})\|^2})^2\|\nabla f(w_{t-1})\|^2 \\ &= \|w_{t-1} - w_*\|^2 - \frac{(f(w_{t-1}) - f^*)^2}{\|\nabla f(w_{t-1})\|^2}\end{aligned}$$

Thus we guarantee a decrease in the iterate distance on each iteration of the algorithm. Now since f is μ-strongly convex and glocally $\{L, L_*\, \delta\}$-smooth (and therefore locally L_*-smooth):

$$\begin{aligned}\|w_t - w_*\|^2 &\leq \|w_{t-1} - w_*\|^2 - \frac{f(w_{t-1}) - f^*}{2L_*} \\ &\leq \|w_{t-1} - w_*\|^2 - \frac{\mu}{4L_*}\|w_{t-1} - w_*\|^2 \\ &= (1 - \frac{\mu}{4L_*})\|w_{t-1} - w_*\|^2\end{aligned}$$

This holds when $f(w_t) - f^* \leq \delta$. Note that the initial iterate in this phase is w_{t_δ}. Let $T = t_\delta + t_\epsilon$. Applying the contraction recursively from $t = t_\delta, \ldots, T$:

$$\|w_T - w_*\|^2 \leq (1 - \frac{\mu}{4L_*})^{t_\epsilon}\|w_{t_\delta} - w_*\|^2$$

Now multiplying both sides by $\frac{L}{2}$, we get that:

$$\frac{L}{2}\|w_T - w_*\|^2 \leq \frac{L}{2}(1 - \frac{\mu}{4L_*})^{t_\epsilon}\|w_{t_\delta} - w_*\|^2$$

Now from Part 1, using (5):

$$\frac{L}{2}\|w_T - w_*\|^2 \leq (1 - \frac{\mu}{4L_*})^{t_\epsilon}\delta$$

Now by (4), if the following holds:

$$\frac{L}{2}\|w_T - w_*\|^2 \leq (1 - \frac{\mu}{4L_*})^{t_\epsilon}\delta \leq \epsilon$$

Then this implies that:

$$f(w_T) - f^* \leq \epsilon$$

as required. This corresponds to an iteration complexity of $t_\epsilon \geq 4(\frac{L_*}{\mu}\log(\frac{\delta}{\epsilon}))$.

Combining both parts, the overall iteration complexity is $T \geq 4(\frac{L}{\mu}\log(\frac{L}{2}\frac{\|w_0 - w_*\|^2}{\delta}) + \frac{L_*}{\mu}\log(\frac{\delta}{\epsilon}))$.
■