Oracle Complexity of Second-Order Methods for Smooth Convex Optimization

Yossi Arjevani yossi.arjevani@weizmann.ac.il
Ohad Shamir ohad.shamir@weizmann.ac.il
Ron Shiff ron.shiff1@gmail.com
Weizmann Institute of Science
Rehovot 7610001, Israel

Abstract

Second-order methods, which utilize gradients as well as Hessians to optimize a given function, are of major importance in mathematical optimization. In this work, we prove tight bounds on the oracle complexity of such methods for smooth convex functions, or equivalently, the worst-case number of iterations required to optimize such functions to a given accuracy. In particular, these bounds indicate when such methods can or cannot improve on gradient-based methods, whose oracle complexity is much better understood. We also provide generalizations of our results to higher-order methods.

1 Introduction

We consider an unconstrained optimization problem of the form

$$\min_{\mathbf{w}\in\mathbb{R}^d} f(\mathbf{w}), \tag{1}$$

where f is a generic smooth and convex function. A natural and fundamental question is how efficiently can we optimize such functions.

We study this question through the well-known framework of oracle complexity [Nemirovsky and Yudin(1983)], which focuses on iterative methods relying on local information. Specifically, it is assumed that the algorithm's access to the function f is limited to an oracle, which given a point $\mathbf{w}$, returns the values and derivatives of the function f at $\mathbf{w}$. This naturally models standard optimization approaches to unstructured problems such as (1), and allows one to study their efficiency, by bounding the number of oracle calls required to reach a given optimization error. Different classes of methods can be distinguished by the type of oracle they use. For example, gradient-based methods (such as gradient descent or accelerated gradient descent) rely on a first-order oracle, which returns gradients, whereas methods such as the Newton method rely on a second-order oracle, which returns gradients as well as Hessians.

The theory of *first-order* oracle complexity is quite well developed [Nemirovsky and Yudin(1983), Nesterov(2004), Nemirovski(2005)]. For example, if the dimension is unrestricted, f in (1) has μ_1-Lipschitz gradients, and the algorithm makes its first oracle query at a point $\mathbf{w}_1$, then the worst-case number of queries T required to attain a point $\mathbf{w}_T$ satisfying $f(\mathbf{w}_T) - \min_{\mathbf{w}} f(\mathbf{w}) \leq \epsilon$ is

$$\Theta\left(\sqrt{\frac{\mu_1 D^2}{\epsilon}}\right), \tag{2}$$

where D is an upper bound on the distance between $\mathbf{w}_1$ and the nearest minimizer of f. Moreover, if the function f is also λ-strongly convex for some $\lambda > 0$[1], then the oracle complexity bound is

$$\Theta\left(\sqrt{\frac{\mu_1}{\lambda}} \cdot \log\left(\frac{\mu_1 D^2}{\epsilon}\right)\right). \tag{3}$$

Both bounds are achievable using accelerated gradient descent [Nesterov(1983)].

However, these bounds do not capture the attainable performance of *second-order* methods, which rely on gradient as well as Hessian information. This is a central class of optimization methods, including the well-known Newton method and its many variants. Clearly, since these methods rely on Hessians as well as

[1] Assuming f is twice-differentiable, this corresponds to $\nabla^2 f(\mathbf{w}) \succeq \lambda I$ uniformly for all $\mathbf{w}$.

OPTML 2017: 10th NIPS Workshop on *Optimization for Machine Learning* (NIPS 2017).

gradients, their oracle complexity can only be better than first-order methods. On the flip side, the per-iteration computational complexity is generally higher, in order to process the additional Hessian information (especially in high-dimensional problems where the Hessian matrix may be very large). Thus, it is natural to ask how much does this added per-iteration complexity pay off in terms of oracle complexity.

To answer this question, one needs good oracle complexity lower bounds for second-order methods, which establish the limits of attainable performance using any such algorithm. Perhaps surprisingly, such results do not seem to currently exist in the literature, and clarifying the oracle complexity of such methods was posed as an important open question (see for example [Nesterov(2008)]). The goal of our paper is to address this gap.

Specifically, we prove [2] that when the dimension is sufficiently large, for the class of convex functions with μ_1-Lipschitz gradients and μ_2-Lipschitz Hessians, the worst-case oracle complexity of any deterministic algorithm is

$$\Omega\left(\min\left\{\sqrt{\frac{\mu_1 D^2}{\epsilon}}\ ,\ \left(\frac{\mu_2 D^3}{\epsilon}\right)^{2/7}\right\}\right). \tag{4}$$

This bound is tight up to constants, as it is matched by a combination of existing methods in the literature (see discussion below). Moreover, if we restrict ourselves to functions which are λ-strongly convex, we prove an oracle complexity lower bound of

$$\Omega\left(\left(\min\left\{\sqrt{\frac{\mu_1}{\lambda}}\ ,\ \left(\frac{\mu_2}{\lambda}D\right)^{2/7}\right\}+\log\log_{18}\left(\frac{\lambda^3/\mu_2^2}{\epsilon}\right)\right)\right). \tag{5}$$

Moreover, we establish that this bound is tight up to logarithmic factors (independent of ϵ), utilizing a novel adaptation of the A-NPE algorithm proposed in [Monteiro and Svaiter(2013)]. These new lower bounds have several implications:

- Perhaps unexpectedly, (5) establishes that one cannot avoid in general a polynomial dependence on geometry-dependent "condition numbers" of the form μ_1/λ or $\mu_2 D/\lambda$, even with second-order methods. This is despite the ability of such methods to favorably alter the geometry of the problem (for example, the Newton method is well-known to be affine invariant).
- To improve on the oracle complexity of first-order methods for strongly-convex problems ((3)) by more than logarithmic factors, one cannot avoid a polynomial dependence on the initial distance D to the optimum. This is despite the fact that the dependence on D with first-order methods is only logarithmic. In fact, when D is sufficiently large (of order $\frac{\mu_1^{7/4}}{\mu_2\lambda^{3/4}}$ or larger), second-order methods cannot improve on the oracle complexity of first-order methods by more than logarithmic factors.
- In the convex case, second-order methods are again no better than first-order methods in certain parameter regimes (i.e., when $\mu_2 \geq \mu_1^{7/4}\sqrt{D}/\epsilon^{3/4}$), despite the availability of more information.

Finally, we show how our proof techniques can be generalized, to establish lower bounds for methods employing higher-order derivatives. In particular, for methods using all derivatives up to order k, we show that for convex functions with μ_k-Lipschitz k-th order derivatives, the oracle complexity is

$$\Omega\left(\left(\frac{\mu_k D^{k+1}}{(k+1)!k\epsilon}\right)^{2/(3k+1)}\right).$$

Note that this directly generalizes (2) for $k=1$, and (4) when $k=2$ and μ_1 is unrestricted.

Related Work

Below, we review some pertinent results in the context of second-order methods.

Perhaps the most well-known and fundamental second-order method is the Newton method, which relies on iterations of the form $\mathbf{w}_{t+1} = \mathbf{w}_t - (\nabla^2 f(\mathbf{w}))^{-1}\nabla f(\mathbf{w})$ (see e.g., [Boyd and Vandenberghe(2004)]). It is well-known that this method exhibits *local* quadratic convergence, in the sense that if f is strictly convex, and the method is initialized close enough to the optimum $\mathbf{w}^* = \arg\min_{\mathbf{w}} f(\mathbf{w})$, then $\mathcal{O}(\log\log(1/\epsilon))$ iterations suffice to reach a solution $\mathbf{w}$ such that $f(\mathbf{w}) - f(\mathbf{w}^*) \leq \epsilon$. However, in order to get global convergence

[2] All statements provided in this workshop paper can be found in the full version.

(starting from an arbitrary point not necessarily close to the optimum), one needs to make some algorithmic modifications, such as introducing a step size parameter or line search, employing trust region methods, or adding various types of regularization (see for example [Conn et al.(2000)] and references therein). Despite the huge literature on the subject, the worst-case global convergence behavior of these methods is not well understood [Nesterov and Polyak(2006)]. For the Newton method with a line search, the number of iterations can be upper bounded by

$$\mathcal{O}\left(\frac{\mu_1^2\mu_2^2}{\lambda^5}(f(\mathbf{w}_1)-f(\mathbf{w}^*))+\log\log_2\left(\frac{\lambda^3/\mu_2^2}{\epsilon}\right)\right),$$

where μ_1,μ_2 are the Lipschitz parameters of the gradients and Hessians respectively, and assuming the function is λ-strongly convex ([Kantorovich(1948)], see also [Boyd and Vandenberghe(2004)]). Note that the first term captures the initial phase required to get sufficiently close to $\mathbf{w}^*$, whereas the second term captures the quadratically convergent phase. Although the final convergence is rapid, the first phase is the dominant one in the bound (unless ϵ is exceedingly small). If f is self-concordant[3], this can be improved to

$$\mathcal{O}\left((f(\mathbf{w}_1)-f(\mathbf{w}^*))+\log\log_2\left(\frac{1}{\epsilon}\right)\right),$$

independent of the strong convexity and Lipschitz parameters ([Nesterov and Nemirovskii(1994)]). Unfortunately, not all practically relevant objective functions are self-concordant. For example, loss functions common in machine learning applications, such as the logistic loss $x\mapsto\log(1+\exp(-x))$, are not self-concordant[4], and our own results utilize the simple but not self-concordant function $x\mapsto|x|^3$.

Returning to our setting of generic convex and smooth functions, and focusing on strongly convex functions for now, the best existing upper bounds (we are aware of) were obtained for cubic-regularized variants of the Newton method, where at each iteration one essentially minimizes a quadratic approximation of the function at the current point, regularized by a cubic term [Nesterov and Polyak(2006), Nesterov(2008)]. The existing analysis (in section 6 of [Nesterov(2008)]) implies an oracle complexity bound of at most

$$\mathcal{O}\left(\left(\frac{\mu_2}{\lambda}D\right)^{1/3}+\log\log_2\left(\frac{\lambda^3/\mu_2^2}{\epsilon}\right)\right),$$

where $D=\|\mathbf{w}_1-\mathbf{w}^*\|$ is the distance from the initialization point $\mathbf{w}_1$ to the optimum $\mathbf{w}^*$ (see section 6 in [Nesterov(2008)], as well as [Cartis et al.(2012)] for another treatment of such cubic-regularized methods). However, we show that a better oracle complexity bound can be obtained, by adapting the A-NPE method proposed in [Monteiro and Svaiter(2013)] and analyzed for convex functions, to the strongly convex case. The resulting complexity upper bound is

$$\mathcal{O}\left(\left(\frac{\mu_2}{\lambda}D\right)^{2/7}\log\left(\frac{\mu_1\mu_2^2D^2}{\lambda^3}\right)+\log\log_2\left(\frac{\lambda^3/\mu_2^2}{\epsilon}\right)\right). \tag{6}$$

An alternative to the above is to use a hybrid scheme, starting with accelerated gradient descent (which is an optimal *first-order* method for strongly convex functions with Lipschitz gradients) and when close enough to the optimal solution, switch to a cubic-regularized Newton method, which is quadratically converging in that region[5]. The required number of iterations is then

$$\mathcal{O}\left(\sqrt{\frac{\mu_1}{\lambda}}\cdot\log\left(\frac{\mu_1\mu_2^2D^2}{\lambda^3}\right)+\log\log_2\left(\frac{\lambda^3/\mu_2^2}{\epsilon}\right)\right), \tag{7}$$

where $D=\|\mathbf{w}_1-\mathbf{w}^*\|$ (see [Nesterov(2004), Nesterov(2008)]). Clearly, by taking the best of (6) and (7) (depending on the parameters), one can theoretically attain an oracle complexity which is the minimum of (6) and (7). This minimum matches (up to a logarithmic factors) the lower bound in (5), which we establish in this paper.

It is interesting to note that the bounds in (6) and (7) are not directly comparable: The first bound has a polynomial dependence on μ_2/λ and $\|\mathbf{w}_1-\mathbf{w}^*\|$, and a logarithmic dependence on μ_1, whereas the second

[3] That is, for any vectors $\mathbf{v},\mathbf{w}$, the function $g(t)=f(\mathbf{w}+t\mathbf{v})$ satisfies $|g'''(t)|\leq 2g''(t)^{3/2}$

[4] These can often be made self-concordant by re-scaling, smoothing and adding regularization (e.g. [Bach(2010)]), but even when possible, these modifications strongly affect the $f(\mathbf{w}_1)-f(\mathbf{w}^*)$ term in the bound, and prevents it from being independent of the strong convexity and Lipschitz parameters.

[5] Instead of cubic-regularized Newton, one can also use the standard Newton method, although the resulting bound using the existing analysis will have slightly worse logarithmic factors.

bound has a polynomial dependence on μ_1/λ, logarithmic dependence on $\|\mathbf{w}_1 - \mathbf{w}^*\|$, and a logarithmic dependence on μ_2. In a rather wide parameter regime (e.g. when D is reasonably large, as often occurs in practice), the bound of the hybrid scheme can be better than that of pure second-order methods. In light of this, [Nesterov(2008)] raised the question of whether second-order schemes are indeed useful at the initial stage of the optimization process, for these types of problems. Our results indicate that indeed, in certain parameter regimes, this is not the case.

Analogous results can be obtained for convex (not necessarily strongly convex) smooth functions. Using an appropriate analysis of the accelerated cubic-regularized Newton method [Nesterov(2008)], one can attain a bound of

$$\mathcal{O}\left(\left(\frac{\mu_2 D^3}{\epsilon}\right)^{1/3}\right) .$$

More recently, [Monteiro and Svaiter(2013)] proposed an accelerated hybrid proximal extragradient method, denoted as A-NPE, which attains a better bound of

$$\mathcal{O}\left(\left(\frac{\mu_2 D^3}{\epsilon}\right)^{2/7}\right) . \tag{8}$$

In addition, using an optimal first-order method (such as accelerated gradient descent), one can attain a bound of

$$\mathcal{O}\left(\sqrt{\frac{\mu_1 D^2}{\epsilon}}\right) . \tag{9}$$

Clearly, by taking the best of the last two approaches (depending on the problem parameters), one can attain an oracle complexity equal to the minimum of the two bounds in (8) and (9). This is matched (up to constants) by the lower bound in (4), which we establish in this paper.

Finally, we discuss the few existing lower bounds known for second-order methods. If μ_2 is not bounded (i.e.,the Hessians are not Lipschitz), it is easy to show that Hessian information is not useful. Specifically, the lower bound of (2) for first-order methods will then also apply to second-order methods, and in fact, to any method based on local information (see [Nemirovsky and Yudin(1983), section 7.2.6] and [Arjevani and Shamir(2016)]). Of course, this lower bound does not apply to second-order methods when μ_2 is bounded. In our setting, it is also possible to prove an $\Omega(\log\log(1/\epsilon))$ lower bound, even in one dimension [Nemirovsky and Yudin(1983), section 8.1.1], but this does not capture the dependence on the strong convexity and Lipschitz parameters. Some algorithm-specific lower bounds in the context of non-convex optimization are provided in [Cartis et al.(2010)]. Finally, we were recently informed of a new work ([Agarwal and Hazan(2017)], yet unpublished at the time of writing), which uses a clean and elegant smoothing approach, to derive second- and higher-order oracle lower bounds directly from known first-order oracle lower bounds, as well as extensions to randomized algorithms. However, the resulting bounds are not as tight as ours.

References

[Agarwal and Hazan(2017)] N. Agarwal and E. Hazan. Lower bounds for higher-order convex optimization. *arXiv preprint arXiv:1710.10329*, 2017.

[Arjevani and Shamir(2016)] Y. Arjevani and O. Shamir. Oracle complexity of second-order methods for finite-sum problems. *arXiv preprint arXiv:1611.04982*, 2016.

[Bach(2010)] F. Bach. Self-concordant analysis for logistic regression. *Electronic Journal of Statistics*, 4: 384–414, 2010.

[Boyd and Vandenberghe(2004)] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[Cartis et al.(2010)] C. Cartis et al. On the complexity of steepest descent, newton's and regularized newton's methods for nonconvex unconstrained optimization problems. *Siam journal on optimization*, 20(6): 2833–2852, 2010.

[Cartis et al.(2012)] C. Cartis et al. Evaluation complexity of adaptive cubic regularization methods for convex unconstrained optimization. *Optimization Methods and Software*, 27(2):197–219, 2012.

[Conn et al.(2000)] A. R. Conn et al. *Trust region methods*. SIAM, 2000.

[Kantorovich(1948)] L. V. Kantorovich. Functional analysis and applied mathematics. *Uspekhi Matematicheskikh Nauk*, 3(6):89–185, 1948.

[Monteiro and Svaiter(2013)] R. D. Monteiro and B. F. Svaiter. An accelerated hybrid proximal extragradient method for convex optimization and its implications to second-order methods. *SIAM Journal on Optimization*, 23(2):1092–1125, 2013.

[Nemirovski(2005)] A. Nemirovski. Efficient methods in convex programming – lecture notes, 2005.

[Nemirovsky and Yudin(1983)] A. Nemirovsky and D. Yudin. *Problem Complexity and Method Efficiency in Optimization*. Wiley-Interscience, 1983.

[Nesterov(1983)] Y. Nesterov. A method of solving a convex programming problem with convergence rate $O(1/k^2)$. *Soviet Mathematics Doklady*, 27(2):372–376, 1983.

[Nesterov(2004)] Y. Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer, 2004.

[Nesterov(2008)] Y. Nesterov. Accelerating the cubic regularization of newton's method on convex problems. *Mathematical Programming*, 112(1):159–181, 2008.

[Nesterov and Nemirovskii(1994)] Y. Nesterov and A. Nemirovskii. *Interior-point polynomial algorithms in convex programming*. SIAM, 1994.

[Nesterov and Polyak(2006)] Y. Nesterov and B. T. Polyak. Cubic regularization of newton method and its global performance. *Mathematical Programming*, 108(1):177–205, 2006.