ESANN 2024 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 9-11 October 2024, i6doc.com publ., ISBN 978-2-87587-090-2. Available from http://www.i6doc.com/en/.

Convergence analysis of an inexact gradient method on smooth convex functions

Pierre Vernimmen and François Glineur

Université catholique de Louvain - ICTEAM institute
B-1348 Louvain-la-Neuve - Belgium

Abstract. We consider the classical gradient method with constant stepsizes where some error is introduced in the computation of each gradient. More specifically, we assume relative inexactness, in the sense that the norm of the difference between the true gradient and its approximate value is bounded by a certain fraction of the gradient norm. We establish a sublinear convergence rate for this inexact method when applied to smooth convex functions, and illustrate on a logistic regression example.

1 Introduction

Inexact gradient methods. First-order methods, and specifically gradient methods, are widely used to solve large-scale machine learning, due to their low computational cost par iteration and their relatively favourable convergence properties, both in theory and in practice [8].

In this paper, we consider one of the most prominent first-order methods, namely the gradient method with constant stepsize. Our goal is to study how its convergence is affected when the gradient that is used at each iteration is computed inexactly. This situation is encountered in a large variety of situations, such as the use of floating point computations with limited accuracy, dependence on data that is only known approximately, or more generally the time-accuracy tradeoff that is almost always present when the objective function (and its gradient) is obtained through another iterative procedure (e.g. a simulation or another optimization process)[1].

First-order methods relying on an inexact gradient have been studied before, using several distinct notions of inexactness. The approximate gradient introduced in [2] is assumed to differ from the true gradient by some error whose norm is bounded. In [4] another notion of inexact gradient is developed, based on the maximal error incurred by the corresponding quadratic upper bound. In these two cases, the approximation error is not directly related to the scale of the gradient, i.e. error is measured in an *absolute* manner.

In this work, we focus on a *relative* notion of inexactness, where the norm of the difference between the true gradient and its approximate value is bounded by a certain fraction of the gradient norm. This notion was introduced [3], and used to analyze the effect of inexactness when dealing with smooth, strongly convex functions. In this work, we extend this analysis to smooth convex functions

[1]Stochasticity may be viewed as another source of inexactness, but we limit the scope of this work to deterministic methods

ESANN 2024 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 9-11 October 2024, i6doc.com publ., ISBN 978-2-87587-090-2. Available from http://www.i6doc.com/en/.

in the absence of strong convexity. More specifically, we will provide a tight convergence analysis of the following inexact gradient method:

Algorithm 1 Inexact Gradient method with constant (normalized) step size h

1: Given an L-smooth convex function f, a starting iterate x_0 and a stepsize h
2: $k \leftarrow 0$
3: **while** $k <$ max_iterations **do**
4: Let d_k be an approximate gradient with δ relative inexactness, i.e.

$$\|d_k - \nabla f(x_k)\| \leq \delta \|\nabla f(x_k)\| \tag{1}$$

5: Compute the next iterate as $x_{k+1} = x_k - \frac{h}{L} d_k$
6: $k \leftarrow k + 1$
7: **end while**

Convergence rates on smooth convex functions. In the case of strongly convex functions, it was proved in [3, Theorem 5.3] that, when applied to an L-smooth and μ-strongly convex objective (with $\mu > 0$), the above gradient method with constant stepsize h and inexactness $\delta < \frac{2\mu}{L+\mu}$ converges linearly according to

$$\|\nabla f(x_1)\| \leq \big(1 - (1-\delta)\tfrac{\mu}{L} h\big) \|\nabla f(x_0)\|,$$

for h sufficiently small, and that this rate is tight. However, when the function becomes merely convex ($\mu \to 0$), this rate becomes ineffective as the constant tends to one. One must instead consider a different measure for the initial iterate x_0, such as its accuracy. In this paper we will prove results of the following type

$$\tfrac{1}{L} \|\nabla f(x_N)\|^2 \leq \tau_N \big(f(x_0) - f(x_*)\big),$$

where constant τ_N describes the rate of convergence after N iterations. In the exact case ($\delta = 0$), it is well-known that the convergence of τ_N for smooth convex functions is of order $\mathcal{O}(\frac{1}{N})$. More precisely, the following tight rate can be found in [9, Theorem 2.1]:

Theorem 1.1. *Algorithm 1 applied to a convex L-smooth function with a constant stepsize $h \in [0; \frac{3}{2}]$ and an exact gradient ($\delta = 0$), started from iterate x_0, generates iterates satisfying*

$$\tfrac{1}{L} \|\nabla f(x_N)\|^2 \leq \frac{f(x_0) - f(x_*)}{Nh + \frac{1}{2}}$$

The main result in this paper is Theorem 2.3, which extends the above to situations where a gradient with relative inexactness is used.

Performance estimation. The two tight convergence rates described above were obtained using the performance estimation framework, as introduced in [5] and further developed in [10], which aims at computing the worst-case convergence rate of some optimization method as the solution of an optimization problem itself. More precisely, the convergence rate τ_N after N iterations of a given algorithm $\mathcal{A}$ over a given class of objective functions $\mathcal{F}$ is equal to the optimal value of the following Performance Estimation Problem (PEP) which

ESANN 2024 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 9-11 October 2024, i6doc.com publ., ISBN 978-2-87587-090-2. Available from http://www.i6doc.com/en/.

can be formulated (with set $I = \{0, 1, \cdots, N, *\}$) and solved numerically as a semidefinite optimization problem (see e.g. [7]):

$$\begin{aligned}\tau_N = \max_{\{x_i, g_i, f_i\}_{i\in I}} \quad & \|g_N\|^2 \\ & x_1, \ldots, x_N \text{ are generated from } x_0 \text{ by algorithm } \mathcal{A}, \\ & \text{there exists an interpolating function } f \in \mathcal{F} \text{ such} \\ & \text{that } f(x_i) = f_i \text{ and } \nabla f(x_i) = g_i \text{ for all } i \in I, \\ & g_* = 0, \text{ and } f_0 - f_* \leq 1.\end{aligned}$$

In order to solve this problem, the condition that links the iterates, gradients and function values (i.e. the variables $\{x_i, g_i, f_i\}_{i\in I}$) to some interpolating function $f \in \mathcal{F}$ must be made explicit. For the class $\mathcal{F}$ we study, namely L-smooth convex functions, it was shown in [10] that this is equivalent to requiring the following inequality, called *interpolation condition*

$$Q_{ij}: \quad f_i \geq f_j + \langle g_j, x_i - x_j\rangle + \tfrac{1}{2L}\|g_j - g_i\|^2 \tag{2}$$

to hold for every pair of indices $i, j \in I$. In addition to allowing to solve the above problem computing the exact value of rate τ_N, one can also obtain a mathematical proof of that guarantee by combining those interpolation inequalities with well-chosen multipliers (which can also be identified as dual variables [10]).

This PEP technique was used to find all results presented in this paper (both theoretical and numerical). The concept of a gradient with δ relative inexactness is obtained by adding (1) to the PEP, rewritten as the convex quadratic constraint $\|d_i - g_i\|^2 \leq \delta^2\|g_i\|^2$ with quantities d_is as additional variables.

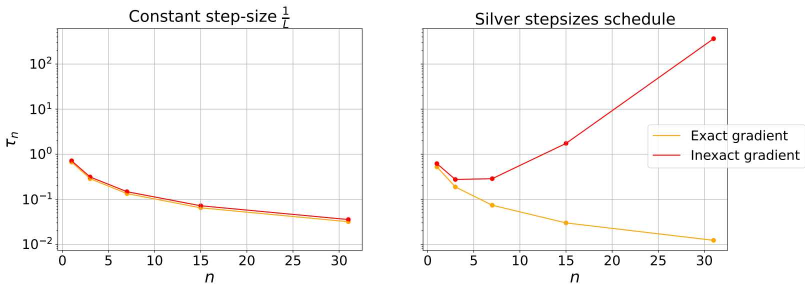

Fig. 1: Rate τ_n as a function of the number n of iterations. Inexactness $\delta = 0.1$.

As an example, we plot on Figure 1 numerical results showing that the behaviour of a first-order method can be very different when relying on an inexact gradient. On the left, the behaviour of the gradient method with constant stepsize $h = 1$ is barely modified. On the right the gradient method with the recently proposed silver stepsize schedule [1], which accelerates convergence in the exact case, deteriorates it heavily in the inexact case.

2 Convergence rate of constant stepsize inexact gradient

In this section, we establish the rate of convergence of the inexact gradient method when using any constant stepsize h such that $h \in [0; \frac{3}{2(1-\delta)}]$. We start with the analysis of a single step.

ESANN 2024 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 9-11 October 2024, i6doc.com publ., ISBN 978-2-87587-090-2. Available from http://www.i6doc.com/en/.

Theorem 2.1 (Rate of convergence of one iteration of inexact gradient descent)**.** *Algorithm 1 applied to a convex L-smooth function with a constant stepsize $h \in]0; \frac{3}{2(1+\delta)}]$ and relative inexactness $\delta \in]0;1[$, started from iterate x_0, generates x_1 satisfying*

$$\tfrac{1}{L}\|\nabla f(x_1)\|^2 \leq \frac{f(x_0)-f(x_1)}{h(1-\delta)} \quad \textit{and} \quad \tfrac{1}{L}\|\nabla f(x_1)\|^2 \leq \frac{f(x_0)-f(x_*)}{\frac{1}{2}+h(1-\delta)}.$$

Proof. The result will be proved without loss of generality for $L=1$. Indeed, if the function is multiplied by some factor c, then its gradient and the smoothness constant L will also be multiplied by c. Thanks to the normalized stepsize $\frac{h}{L}$ in Algorithm 1, iterates will stay the same, hence their function values will be multiplied by c, and the squared norms of their gradients by c^2, which preserves both displayed rates thanks to the $\frac{1}{L}$ factor in front of the left-hand sides.

To proceed, we introduce the notation $D = d_0 - g_0$ and sum the two interpolation conditions Q_{01} and Q_{10} with multipliers $\lambda_{01} = 2$ and $\lambda_{10} = 1$ to inequality $\|d_0 - g_0\|^2 \leq \delta^2 \| g_0\|^2$ with multiplier $\frac{h}{2\delta}$, which gives

$$f_0 - f_1 \geq \left(\tfrac{3}{2} - h - \tfrac{h\delta}{2}\right)\|g_0\|^2 + \tfrac{3}{2}\|g_1\|^2 + \tfrac{h}{2\delta}\|D\|^2 + (2h-3)\langle g_0, g_1\rangle + 2h\langle g_1, D\rangle - h\langle g_0, D\rangle,$$

which can be rewritten as $f_0 - f_1 \geq (g_0, g_1, D)^T M (g_0, g_1, D)$ with the following matrix M:

$$M = \begin{pmatrix} (3/2 - h - h\delta/2) & (h-3/2) & -h/2 \\ (h-3/2) & (3/2 - h(1-\delta)) & h \\ -h/2 & h & h/2\delta \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 \\ 0 & h(1-\delta) & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

If we can show that first summand in matrix M, which we will call P, is positive semidefinite, then it will imply that $f_0 - f_1 \geq h(1-\delta)\|g_1\|^2$, which is the first rate we want to prove. To do so, we partition P in four blocks as $P = \begin{pmatrix} A & B \\ B^T & C \end{pmatrix}$ with

$$A = \begin{pmatrix} \frac{3}{2} - h - \frac{h\delta}{2} & -\frac{3}{2} + h \\ -\frac{3}{2} + h & \frac{3}{2} - h(1+\delta) \end{pmatrix}, \quad B = \begin{pmatrix} \frac{-h}{2} \\ h \end{pmatrix}, \quad C = \frac{h}{2\delta}.$$

Since C is positive definite, matrix P is positive semidefinite if and only if the Schur complement $A - BC^{-1}B^T$ is also positive semidefinite, which is equal to

$$\begin{pmatrix} \frac{3}{2} - h - \frac{h\delta}{2} & -\frac{3}{2} + h \\ -\frac{3}{2} + h & \frac{3}{2} - h(1+\delta) \end{pmatrix} - \frac{2\delta}{h}\begin{pmatrix} \frac{h^2}{4} & -\frac{h^2}{2} \\ -\frac{h^2}{2} & h^2 \end{pmatrix} = \left(\tfrac{3}{2} - h(1+\delta)\right)\begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}.$$

This is clearly positive semidefinite if and only if $\frac{3}{2} - h(1+\delta) \geq 0$, which is guaranteed by the hypothesis $h \leq \frac{3}{2(1+\delta)}$. To show the second part of the theorem, we use another interpolation inequality (known as the descent lemma)

$$Q_{1*}: f_1 \geq f_* + \langle g_*, f_1 - f_*\rangle + \tfrac{1}{2}\|g_* - g_1\|^2 = f_* + \tfrac{1}{2}\|g_1\|^2,$$

and add it to $f_0 - f_1 \geq h(1-\delta)\|g_1\|^2$ to obtain the desired rate $f_0 - f_1 \geq \left(\frac{1}{2} + h(1-\delta)\right)\|g_1\|^2$.

□

ESANN 2024 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 9-11 October 2024, i6doc.com publ., ISBN 978-2-87587-090-2. Available from http://www.i6doc.com/en/.

Let us now show that these rates cannot be improved.

Theorem 2.2. *Both rates proven in Theorem 2.1 are tight.*

Proof. Again we consider the case $L = 1$ without loss of generality. The following univariate function, often called Huber function, is 1-smooth

$$f(x) = \frac{1}{\sqrt{h(1-\delta)}}|x| - \frac{1}{2h(1-\delta)} \text{ when } |x| \geq \frac{1}{\sqrt{h(1-\delta)}}, \text{ otherwise } f(x) = \frac{1}{2}x^2.$$

Assuming that $x_0 \geq \frac{1}{\sqrt{h(1-\delta)}}$, we have that $\nabla f(x) = \frac{1}{\sqrt{h(1-\delta)}}$. If we now choose $d_0 = (1-\delta)\nabla f(x_0)$, which clearly satisfies the definition of δ inexactness, iterate x_1 will become $x_1 = x_0 - \sqrt{h(1-\delta)}$. Now choosing x_0 such that $x_1 = \frac{1}{\sqrt{h(1-\delta)}}$, i.e. $x_0 = \sqrt{h(1-\delta)} + \frac{1}{\sqrt{h(1-\delta)}}$, it is straightforward to check that $f(x_0) - f(x_1) = 1$ and we have shown that $\|g_1\|^2 = \frac{f(x_0)-f(x_1)}{h(1-\delta)}$. The second rate can be checked to be tight in a similar way, using the same f, x_0, d_0, x_1 and $f_* = 0$. □

To conclude, we derive a convergence rate characterizing the behaviour of the inexact gradient method after several iterations. We first note that, somehow surprisingly, monotonicity of the gradient cannot be guaranteed for $\delta > 0$, i.e. $\|g_{i+1}\|^2 \leq \|g_i\|^2$ is not true in general for iterates generated by Algorithm 1. However, a bound on the minimum norm among all gradients can still be derived.

Theorem 2.3. *Algorithm 1 applied to a convex L-smooth function with a constant stepsize $h \in]0; \frac{3}{2(1+\delta)}]$ and relative inexactness $\delta \in]0; 1[$, started from iterate x_0, generates iterates satisfying*

$$\tfrac{1}{L} \min_{k \in \{1, \cdots, N\}} \|\nabla f(x_k)\|^2 \leq \frac{f(x_0) - f(x_*)}{\frac{1}{2} + Nh(1-\delta)}.$$

Proof. Writing the first rate in Theorem 2.1 on consecutive iterates (x_k, x_{k+1}) gives $h(1-\delta)\|\nabla f(x_{k+1})\|^2 \leq f(x_k) - f(x_{k+1})$. Summing from $k = 0$ to $N - 1$, and adding the descent lemma $f(x_N) - f(x_*) \geq \frac{1}{2}\|\nabla f(x_N)\|^2$ as in the last step in the proof of Theorem 2.1 gives, after telescoping the right-hand side

$$\tfrac{1}{2}\|\nabla f(x_N)\|^2 + h(1-\delta)\textstyle\sum_{k=0}^{N-1} \|\nabla f(x_{k+1})\|^2 \leq f(x_0) - f(x_*).$$

Using the fact that every squared gradient in the left-hand side is greater than the one with the minimum value among them gives the sought inequality

$$\left(\tfrac{1}{2} + Nh(1-\delta)\right) \min_{k \in \{1, \cdots, N\}} \|\nabla f(x_k)\|^2 \leq f(x_0) - f(x_*).$$

□

Hence Algorithm 1 will converge for any $\delta \in]0; 1[$, with nearly the same sublinear convergence rate as in the exact case except for an extra $(1 - \delta)$ factor.

ESANN 2024 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 9-11 October 2024, i6doc.com publ., ISBN 978-2-87587-090-2. Available from http://www.i6doc.com/en/.

3 Numerical experiment

To conclude, we perform a numerical experiment to validate our theoretical findings. Given some initial point, we apply one step of both the exact and inexact gradient methods to minimize the objective of a logistic regression problem for the standard iris dataset ([6]). For the latter method, we choose $\delta = 0.15$ and, as our focus is on worst-case analysis, we pick the worst possible direction satisfying the inexactness condition. More precisely, we add to the true gradient an error term (with maximum norm) going opposite to the minimizer, i.e. add a multiple of $-(x_0 - x_*)$. Figure 2 shows one example of the observed numerical rates of convergence τ_1 for some range of the stepsize parameter h. This confirms that convergence is slower with the inexact gradient, and that the optimal stepsize for the inexact gradient is shorter than for the exact gradient (echoing $\frac{3}{2}$ vs. $\frac{3}{2(1+\delta)}$). However, as opposed to the theoretical worst-case, longer step sizes notably affect inexact gradients more than exact ones, as seen when h increases.

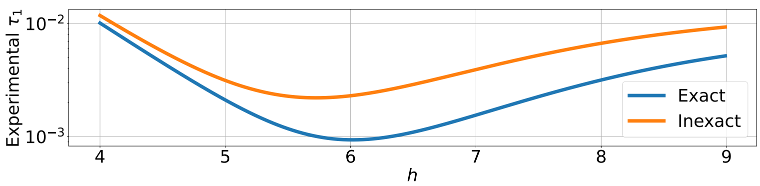

Fig. 2: Experimental rate of convergence $\tau_1 = \frac{\|g_1\|^2}{f_0 - f_1}$ after one iteration.

References

[1] Jason M Altschuler and Pablo A Parrilo. Acceleration by stepsize hedging ii: Silver stepsize schedule for smooth convex optimization. *arXiv:2309.16530*, 2023.

[2] Alexandre d'Aspremont. Smooth optimization with approximate gradient. *SIAM Journal on Optimization*, 19(3):1171–1183, 2008.

[3] Etienne De Klerk, Francois Glineur, and Adrien B Taylor. Worst-case convergence analysis of inexact gradient and newton methods through semidefinite programming performance estimation. *SIAM Journal on Optimization*, 30(3):2053–2082, 2020.

[4] Olivier Devolder, François Glineur, and Yurii Nesterov. First-order methods of smooth convex optimization with inexact oracle. *Mathematical Programming*, 146:37–75, 2014.

[5] Yoel Drori and Marc Teboulle. Performance of first-order methods for smooth convex minimization: a novel approach. *Mathematical Programming*, 145(1-2):451–482, 2014.

[6] Ronald A Fisher. The use of multiple measurements in taxonomic problems. *Annals of eugenics*, 7(2):179–188, 1936.

[7] Baptiste Goujaud, Céline Moucer, François Glineur, Julien Hendrickx, Adrien Taylor, and Aymeric Dieuleveut. PEPit: computer-assisted worst-case analyses of first-order optimization methods in python. *arXiv:2201.04040*, 2022.

[8] Guanghui Lan. *First-order and stochastic optimization methods for machine learning*, volume 1. Springer.

[9] Teodor Rotaru, François Glineur, and Panagiotis Patrinos. Exact worst-case convergence rates of gradient descent: a complete analysis for all constant stepsizes over nonconvex and convex functions. *arXiv:2406.17506*, 2024.

[10] Adrien B Taylor, Julien M Hendrickx, and François Glineur. Smooth strongly convex interpolation and exact worst-case performance of first-order methods. *Mathematical Programming*, 161:307–345, 2017.