# Convergence Rates of Inexact Proximal-Gradient Methods for Convex Optimization

**Mark Schmidt** mark.schmidt@inria.fr
**Nicolas Le Roux** nicolas@le-roux.name
**Francis Bach** francis.bach@ens.fr

INRIA - SIERRA Project Team
École Normale Supérieure, Paris

## Abstract

We consider the problem of optimizing the sum of a smooth convex function and a non-smooth convex function using proximal-gradient methods, where an error is present in the calculation of the gradient of the smooth term or in the proximity operator with respect to the non-smooth term. We show that both the basic proximal-gradient method and the accelerated proximal-gradient method achieve the same convergence rate as in the error-free case, provided that the errors decrease at appropriate rates. Using these rates, we perform as well as or better than a carefully chosen fixed error level on a set of structured sparsity problems.

## 1 Introduction

In recent years the importance of taking advantage of the structure of convex optimization problems has become a topic of intense research in the machine learning community. This is particularly true of techniques for non-smooth optimization, where taking advantage of the structure of non-smooth terms seems to be crucial to obtaining good performance. Proximal-gradient methods and *accelerated* proximal-gradient methods [1, 2] are among the most important methods for taking advantage of the structure of many of the non-smooth optimization problems that arise in practice. In particular, these methods address composite optimization problems of the form

$$\underset{x\in\mathbb{R}^d}{\text{minimize}} \quad f(x) := g(x) + h(x), \tag{1}$$

where $g$ and $h$ are convex functions but only $g$ is smooth. One of the most well-studied instances of this type of problem is $\ell_1$-regularized least squares [3, 4],

$$\underset{x\in\mathbb{R}^d}{\text{minimize}} \ \frac{1}{2}\|Ax-b\|^2 + \lambda\|x\|_1,$$

where we use $\|\cdot\|$ to denote the standard $\ell_2$-norm.

Proximal-gradient methods are an appealing approach for solving these types of non-smooth optimization problems because of their fast theoretical convergence rates and strong practical performance. While classical subgradient methods only achieve an error level on the objective function of $O(1/\sqrt{k})$ after $k$ iterations, proximal-gradient methods have an error of $O(1/k)$ while *accelerated* proximal-gradient methods futher reduce this to $O(1/k^2)$ [1, 2]. That is, accelerated proximal-gradient methods for *non-smooth* convex optimization achieve the same optimal convergence rate that accelerated gradient methods achieve for *smooth* optimization.

Each iteration of a proximal-gradient method requires the calculation of the proximity operator,

$$\text{prox}_L(y) = \underset{x\in\mathbb{R}^d}{\arg\min} \ \frac{L}{2}\|x-y\|^2 + h(x), \tag{2}$$

where $L$ is the Lipschitz constant of the gradient of $g$. We can efficiently compute an analytic solution to this problem for several notable choices of $h$, including the case of $\ell_1$-regularization and disjoint group $\ell_1$-regularization [5, 6]. However, in many scenarios the proximity operator may not have an analytic solution, or it may be very expensive to compute this solution exactly. This includes important problems such as total-variation regularization and its generalizations like the graph-guided fused-LASSO [7, 8], nuclear-norm regularization and other regularizers on the singular values of matrices [9, 10], and different formulations of overlapping group $\ell_1$-regularization with general groups [11, 12]. Despite the difficulty in computing the exact proximity operator for these regularizers, efficient methods have been developed to compute *approximate* proximity operators in all of these cases; accelerated projected gradient and Newton-like methods that work with a smooth dual problem have been used to compute approximate proximity operators in the context of total-variation regularization [7, 13], Krylov subspace methods and low-rank representations have been used to compute approximate proximity operators in the context of nuclear-norm regularization [9, 10], and variants of Dykstra's algorithm (and related dual methods) have been used to compute approximate proximity operators in the context of overlapping group $\ell_1$-regularization [12, 14, 15].

It is known that proximal-gradient methods that use an approximate proximity operator converge under only weak assumptions [16, 17]; we briefly review this and other related work in the next section. However, despite the many recent works showing impressive empirical performance of (accelerated) proximal-gradient methods that use an approximate proximity operator [7, 13, 9, 10, 14, 15], up until recently there was no theoretical analysis on how the error in the calculation of the proximity operator affects the convergence *rate* of proximal-gradient methods. In this work we show in several contexts that, provided the error in the proximity operator calculation is controlled in an appropriate way, inexact proximal-gradient strategies achieve the *same* convergence rates as the corresponding exact methods. In particular, in Section 4 we first consider convex objectives and analyze the inexact proximal-gradient (Proposition 1) and accelerated proximal-gradient (Proposition 2) methods. We then analyze these two algorithms for strongly convex objectives (Proposition 3 and Proposition 4). Note that in these analyses, we also consider the possibility that there is an error in the calculation of the gradient of $g$. We then present an experimental comparison of various inexact proximal-gradient strategies in the context of solving a structured sparsity problem (Section 5).

## 2 Related Work

The algorithm we shall focus on in this paper is the proximal-gradient method

$$x_k = \text{prox}_L\left[y_{k-1} - (1/L)(g'(y_{k-1}) + e_k)\right] \ , \tag{3}$$

where $e_k$ is the error in the calculation of the gradient and the proximity problem (2) is solved inexactly so that $x_k$ has an error of $\varepsilon_k$ in terms of the proximal objective function (2). In the basic proximal-gradient method we choose $y_k = x_k$, while in the accelerated proximal-gradient method we choose $y_k = x_k + \beta_k(x_k - x_{k-1})$, where the sequence $(\beta_k)$ is chosen appropriately.

There is a substantial amount of work on methods that use an exact proximity operator but have an error in the gradient calculation, corresponding to the special case where $\varepsilon_k = 0$ but $e_k$ is non-zero. For example, when the $e_k$ are independent, zero-mean, and finite-variance random variables, then proximal-gradient methods achieve the (optimal) error level of $O(1/\sqrt{k})$ [18, 19]. This is different than the scenario we analyze in this paper since we do *not* assume unbiased nor independent errors, but instead consider a sequence of errors converging to 0. This leads to faster convergence rates, and makes our analysis applicable to the case of deterministic (and even adversarial) errors.

Several authors have recently analyzed the case of a fixed deterministic error in the gradient, and shown that accelerated gradient methods achieve the optimal convergence rate up to some accuracy that depends on the fixed error level [20, 21, 22], while the earlier work of [23] analyzes the gradient method in the context of a fixed error level. This contrasts with our analysis, where by allowing the error to change at every iteration we can achieve convergence to the optimal solution. Also, we can tolerate a large error in early iterations when we are far from the solution, which may lead to substantial computational gains. Other authors have analyzed the convergence rate of the gradient and projected-gradient methods with a decreasing sequence of errors [24, 25], but this analysis does not consider the important class of accelerated gradient methods. In contrast, the analysis of [22] allows a decreasing sequence of errors (though convergence rates in this context are not explicitly

mentioned) and considers the accelerated projected-gradient method. However, the authors of this work only consider the case of an exact projection step, and they assume the availability of an oracle that yields global lower and upper bounds on the function. This non-intuitive oracle leads to a novel analysis of smoothing methods, but leads to slower convergence rates than proximal-gradient methods. The analysis of [21] considers errors in both the gradient and projection operators for accelerated projected-gradient methods, but this analysis requires that the domain of the function is compact. None of these works consider proximal-gradient methods.

In the context of *proximal-point* algorithms, there is a substantial literature on using inexact proximity operators with a decreasing sequence of errors, dating back to the seminal work of Rockafeller [26]. Accelerated proximal-point methods with a decreasing sequence of errors have also been examined, beginning with [27]. However, unlike proximal-gradient methods where the proximity operator is only computed with respect to the non-smooth function $h$, proximal-point methods require the calculation of the proximity operator with respect to the full objective function. In the context of composite optimization problems of the form (1), this requires the calculation of the proximity operator with respect to $g + h$. Since it ignores the structure of the problem, this proximity operator may be as difficult to compute (even approximately) as the minimizer of the original problem.

Convergence of inexact proximal-gradient methods can be established with only weak assumptions on the method used to approximately solve (2). For example, we can establish that inexact proximal-gradient methods converge under some closedness assumptions on the mapping induced by the approximate proximity operator, and the assumption that the algorithm used to compute the inexact proximity operator achieves sufficient descent on problem (2) compared to the previous iteration $x_{k-1}$ [16]. Convergence of inexact proximal-gradient methods can also be established under the assumption that the norms of the errors are summable [17]. However, these prior works did not consider the *rate* of convergence of inexact proximal-gradient methods, nor did they consider accelerated proximal-gradient methods. Indeed, the authors of [7] chose to use the non-accelerated variant of the proximal-gradient algorithm since even convergence of the accelerated proximal-gradient method had not been established under an inexact proximity operator.

While preparing the final version of this work, [28] independently gave an analysis of the accelerated proximal-gradient method with an inexact proximity operator and a decreasing sequence of errors (assuming an exact gradient). Further, their analysis leads to a weaker dependence on the errors than in our Proposition 2. However, while we only assume that the proximal problem can be solved up to a certain accuracy, they make the much stronger assumption that the inexact proximity operator yields an $\varepsilon_k$-subdifferential of $h$ [28, Definition 2.1]. Our analysis can be modified to give an improved dependence on the errors under this stronger assumption. In particular, the terms in $\sqrt{\varepsilon_i}$ disappear from the expressions of $A_k$, $\widetilde{A}_k$ and $\widehat{A}_k$ appearing in the propositions, leading to the optimal convergence rate with a slower decay of $\varepsilon_i$. More details may be found in [29].

## 3 Notation and Assumptions

In this work, we assume that the smooth function $g$ in (1) is convex and differentiable, and that its gradient $g'$ is Lipschitz-continuous with constant $L$, meaning that for all $x$ and $y$ in $\mathbb{R}^d$ we have

$$\|g'(x) - g'(y)\| \leqslant L\|x - y\| \ .$$

This is a standard assumption in differentiable optimization, see [30, §2.1.1]. If $g$ is twice-differentiable, this corresponds to the assumption that the eigenvalues of its Hessian are bounded above by $L$. In Propositions 3 and 4 only, we will also assume that $g$ is $\mu$-strongly convex (see [30, §2.1.3]), meaning that for all $x$ and $y$ in $\mathbb{R}^d$ we have

$$g(y) \geqslant g(x) + \langle g'(x), y - x\rangle + \frac{\mu}{2}||y - x||^2.$$

In contrast to these assumptions on $g$, we will only assume that $h$ in (1) is a lower semi-continuous proper convex function (see [31, §1.2]), but will not assume that $h$ is differentiable or Lipschitz-continuous. This allows $h$ to be any real-valued convex function, but also allows for the possibility that $h$ is an extended real-valued convex function. For example, $h$ could be the indicator function of a convex set, and in this case the proximity operator becomes the projection operator.

We will use $x_k$ to denote the parameter vector at iteration $k$, and $x^*$ to denote a minimizer of $f$. We assume that such an $x^*$ exists, but do not assume that it is unique. We use $e_k$ to denote the error in the calculation of the gradient at iteration $k$, and we use $\varepsilon_k$ to denote the error in the proximal objective function achieved by $x_k$, meaning that

$$\frac{L}{2}\|x_k - y\|^2 + h(x_k) \leqslant \varepsilon_k + \min_{x \in \mathbb{R}^d} \left\{ \frac{L}{2}\|x - y\|^2 + h(x) \right\}, \tag{4}$$

where $y = y_{k-1} - (1/L)(g'(y_{k-1}) + e_k))$. Note that the proximal optimization problem (2) is strongly convex and in practice we are often able to obtain such bounds via a duality gap (e.g., see [12] for the case of overlapping group $\ell_1$-regularization).

## 4 Convergence Rates of Inexact Proximal-Gradient Methods

In this section we present the analysis of the convergence rates of inexact proximal-gradient methods as a function of the sequences of solution accuracies to the proximal problems $(\varepsilon_k)$, and the sequences of magnitudes of the errors in the gradient calculations $(\|e_k\|)$. We shall use (**H**) to denote the set of four assumptions which will be made for each proposition:

- $g$ is convex and has $L$-Lipschitz-continuous gradient;
- $h$ is a lower semi-continuous proper convex function;
- The function $f = g + h$ attains its minimum at a certain $x^* \in \mathbb{R}^n$;
- $x_k$ is an $\varepsilon_k$-optimal solution to the proximal problem (2) in the sense of (4).

We first consider the basic proximal-gradient method in the convex case:

**Proposition 1 (Basic proximal-gradient method - Convexity)** *Assume (**H**) and that we iterate recursion* (3) *with $y_k = x_k$. Then, for all $k \geqslant 1$, we have*

$$f\left(\frac{1}{k}\sum_{i=1}^{k} x_i\right) - f(x^*) \leqslant \frac{L}{2k}\left(\|x_0 - x^*\| + 2A_k + \sqrt{2B_k}\right)^2, \tag{5}$$

$$\text{with } A_k = \sum_{i=1}^{k}\left(\frac{\|e_i\|}{L} + \sqrt{\frac{2\varepsilon_i}{L}}\right), \quad B_k = \sum_{i=1}^{k}\frac{\varepsilon_i}{L}.$$

The proof may be found in [29]. Note that while we have stated the proposition in terms of the function value achieved by the average of the iterates, it trivially also holds for the iteration that achieves the lowest function value. This result implies that the well-known $O(1/k)$ convergence rate for the gradient method without errors *still holds* when both $(\|e_k\|)$ and $(\sqrt{\varepsilon_k})$ are summable. A sufficient condition to achieve this is that $\|e_k\|$ decreases as $O(1/k^{1+\delta})$ while $\varepsilon_k$ decreases as $O(1/k^{2+\delta'})$ for any $\delta, \delta' > 0$. Note that a faster convergence of these two errors will not improve the convergence rate, but will yield a better constant factor.

It is interesting to consider what happens if $(\|e_k\|)$ or $(\sqrt{\varepsilon_k})$ is not summable. For instance, if $\|e_k\|$ and $\sqrt{\varepsilon_k}$ decrease as $O(1/k)$, then $A_k$ grows as $O(\log k)$ (note that $B_k$ is always smaller than $A_k$) and the convergence of the function values is in $O\left(\frac{\log^2 k}{k}\right)$. Finally, a necessary condition to obtain convergence is that the partial sums $A_k$ and $B_k$ need to be in $o(\sqrt{k})$.

We now turn to the case of an accelerated proximal-gradient method. We focus on a basic variant of the algorithm where $\beta_k$ is set to $(k-1)/(k+2)$ [32, Eq. (19) and (27)]:

**Proposition 2 (Accelerated proximal-gradient method - Convexity)** *Assume (**H**) and that we iterate recursion* (3) *with $y_k = x_k + \frac{k-1}{k+2}(x_k - x_{k-1})$. Then, for all $k \geqslant 1$, we have*

$$f(x_k) - f(x^*) \leqslant \frac{2L}{(k+1)^2}\left(\|x_0 - x^*\| + 2\widetilde{A}_k + \sqrt{2\widetilde{B}_k}\right)^2, \tag{6}$$

$$\text{with } \widetilde{A}_k = \sum_{i=1}^{k} i\left(\frac{\|e_i\|}{L} + \sqrt{\frac{2\varepsilon_i}{L}}\right), \quad \widetilde{B}_k = \sum_{i=1}^{k}\frac{i^2\varepsilon_i}{L}.$$

In this case, we require the series $(k\|e_k\|)$ and $(k\sqrt{\varepsilon_k})$ to be summable to achieve the optimal $O(1/k^2)$ rate, which is an (unsurprisingly) stronger constraint than in the basic case. A sufficient condition is for $\|e_k\|$ and $\sqrt{\varepsilon_k}$ to decrease as $O(1/k^{2+\delta})$ for any $\delta > 0$. Note that, as opposed to Proposition 1 that is stated for the average iterate, this bound is for the last iterate $x_k$.

Again, it is interesting to see what happens when the summability assumption is not met. First, if $\|e_k\|$ or $\sqrt{\varepsilon_k}$ decreases at a rate of $O(1/k^2)$, then $k(\|e_k\| + \sqrt{e_k})$ decreases as $O(1/k)$ and $\widetilde{A}_k$ grows as $O(\log k)$ (note that $\widetilde{B}_k$ is always smaller than $\widetilde{A}_k$), yielding a convergence rate of $O\left(\frac{\log^2 k}{k^2}\right)$ for $f(x_k) - f(x^*)$. Also, and perhaps more interestingly, if $\|e_k\|$ or $\sqrt{\varepsilon_k}$ decreases at a rate of $O(1/k)$, Eq. (6) does not guarantee convergence of the function values. More generally, the form of $\widetilde{A}_k$ and $\widetilde{B}_k$ indicates that errors have a greater effect on the accelerated method than on the basic method. Hence, as also discussed in [22], unlike in the error-free case the accelerated method may not necessarily be better than the basic method because it is more sensitive to errors in the computation.

In the case where $g$ is *strongly* convex it is possible to obtain linear convergence rates that depend on the ratio $\gamma = \mu/L$ as opposed to the sublinear convergence rates discussed above. In particular, we obtain the following convergence rate on the iterates of the basic proximal-gradient method:

**Proposition 3 (Basic proximal-gradient method - Strong convexity)** *Assume (**H**), that $g$ is $\mu$-strongly convex, and that we iterate recursion* (3) *with $y_k = x_k$. Then, for all $k \geqslant 1$, we have:*

$$\|x_k - x^*\| \leqslant (1-\gamma)^k \left(\|x_0 - x^*\| + \bar{A}_k\right), \tag{7}$$

$$\text{with} \quad \bar{A}_k = \sum_{i=1}^{k} (1-\gamma)^{-i} \left(\frac{\|e_i\|}{L} + \sqrt{\frac{2\varepsilon_i}{L}}\right) .$$

A consequence of this proposition is that we obtain a linear rate of convergence even in the presence of errors, provided that $\|e_k\|$ and $\sqrt{\varepsilon_k}$ decrease linearly to 0. If they do so at a rate of $Q' < (1-\gamma)$, then the convergence rate of $\|x_k - x^*\|$ is linear with constant $(1-\gamma)$, as in the error-free algorithm. If we have $Q' > (1-\gamma)$, then the convergence of $\|x_k - x^*\|$ is linear with constant $Q'$. If we have $Q' = (1-\gamma)$, then $\|x_k - x^*\|$ converges to 0 as $O(k\,(1-\gamma)^k) = o\left([(1-\gamma)+\delta']^k\right)$ for all $\delta' > 0$.

Finally, we consider the accelerated proximal-gradient algorithm when $g$ is strongly convex. We focus on a basic variant of the algorithm where $\beta_k$ is set to $(1-\sqrt{\gamma})/(1+\sqrt{\gamma})$ [30, §2.2.1]:

**Proposition 4 (Accelerated proximal-gradient method - Strong convexity)** *Assume (**H**), that $g$ is $\mu$-strongly convex, and that we iterate recursion* (3) *with $y_k = x_k + \frac{1-\sqrt{\gamma}}{1+\sqrt{\gamma}}(x_k - x_{k-1})$. Then, for all $k \geqslant 1$, we have*

$$f(x_k) - f(x^*) \leqslant (1-\sqrt{\gamma})^k \left(\sqrt{2(f(x_0) - f(x^*))} + \widehat{A}_k \sqrt{\frac{2}{\mu}} + \sqrt{\widehat{B}_k}\right)^2, \tag{8}$$

$$\text{with } \widehat{A}_k = \sum_{i=1}^{k} \left(\|e_i\| + \sqrt{2L\varepsilon_i}\right)(1-\sqrt{\gamma})^{-i/2} \quad , \quad \widehat{B}_k = \sum_{i=1}^{k} \varepsilon_i \,(1-\sqrt{\gamma})^{-i} \quad .$$

Note that while we have stated the result in terms of function values, we obtain an analogous result on the iterates because by strong convexity of $f$ we have

$$\frac{\mu}{2}||x_k - x_*||^2 \leq f(x_k) - f(x_*).$$

This proposition implies that we obtain a linear rate of convergence in the presence of errors provided that $||e_k||^2$ and $\varepsilon_k$ decrease linearly to 0. If they do so at a rate $Q' < (1-\sqrt{\gamma})$, then the constant is $(1-\sqrt{\gamma})$, while if $Q' > (1-\sqrt{\gamma})$ then the constant will be $Q'$. Thus, the accelerated inexact proximal-gradient method will have a faster convergence rate than the *exact* basic proximal-gradient method provided that $Q' < (1-\gamma)$. Oddly, in our analysis of the strongly convex case, the accelerated method is *less sensitive* to errors than the basic method. However, unlike the basic method, the accelerated method requires knowing $\mu$ in addition to $L$. If $\mu$ is misspecified, then the convergence rate of the accelerated method may be slower than the basic method.

## 5 Experiments

We tested the basic inexact proximal-gradient and accelerated proximal-gradient methods on the CUR-like factorization optimization problem introduced in [33] to approximate a given matrix $W$,

$$\min_X \tfrac{1}{2}\|W - WXW\|_F^2 + \lambda_{\text{row}} \sum_{i=1}^{n_r} ||X^i||_p + \lambda_{col} \sum_{j=1}^{n_c} ||X_j||_p \ .$$

Under an appropriate choice of $p$, this optimization problem yields a matrix $X$ with sparse rows *and* sparse columns, meaning that entire rows and columns of the matrix $X$ are set to exactly zero. In [33], the authors used an accelerated proximal-gradient method and chose $p = \infty$ since under this choice the proximity operator can be computed exactly. However, this has the undesirable effect that it also encourages all values in the same row (or column) to have the same magnitude. The more natural choice of $p = 2$ was not explored since in this case there is no known algorithm to exactly compute the proximity operator.

Our experiments focused on the case of $p = 2$. In this case, it is possible to very quickly compute an approximate proximity operator using the block coordinate descent (BCD) algorithm presented in [12], which is equivalent to the proximal variant of Dykstra's algorithm introduced by [34]. In our implementation of the BCD method, we alternate between computing the proximity operator with respect to the rows and to the columns. Since the BCD method allows us to compute a duality gap when solving the proximal problem, we can run the method until the duality gap is below a given error threshold $\varepsilon_k$ to find an $x_{k+1}$ satisfying (4).

In our experiments, we used the four data sets examined by [33][1] and we choose $\lambda_{row} = .01$ and $\lambda_{col} = .01$, which yielded approximately 25–40% non-zero entries in $X$ (depending on the data set). Rather than assuming we are given the Lipschitz constant $L$, on the first iteration we set $L$ to 1 and following [2] we double our estimate anytime $g(x_k) > g(y_{k-1}) + \langle g'(y_{k-1}), x_k - y_{k-1}\rangle + (L/2)||x_k - y_{k-1}||^2$. We tested three different ways to terminate the approximate proximal problem, each parameterized by a parameter $\alpha$:

- $\varepsilon_k = 1/k^\alpha$: Running the BCD algorithm until the duality gap is below $1/k^\alpha$.
- $\varepsilon_k = \alpha$: Running the BCD algorithm until the duality gap is below $\alpha$.
- $n = \alpha$: Running the BCD algorithm for a fixed number of iterations $\alpha$.

Note that all three strategies lead to global convergence in the case of the basic proximal-gradient method, the first two give a convergence rate up to some fixed optimality tolerance, and in this paper we have shown that the first one (for large enough $\alpha$) yields a convergence rate for an arbitrary optimality tolerance. Note that the iterates produced by the BCD iterations are *sparse*, so we expected the algorithms to spend the majority of their time solving the proximity problem. Thus, we used the function value against the number of BCD iterations as a measure of performance. We plot the results after 500 BCD iterations for the first two data sets for the proximal-gradient method in Figure 1, and the accelerated proximal-gradient method in Figure 2. The results for the other two data sets are similar, and are included in [29]. In these plots, the first column varies $\alpha$ using the choice $\varepsilon_k = 1/k^\alpha$, the second column varies $\alpha$ using the choice $\varepsilon_k = \alpha$, and the third column varies $\alpha$ using the choice $n = \alpha$. We also include one of the best methods from the first column in the second and third columns as a reference.

In the context of proximal-gradient methods the choice of $\varepsilon_k = 1/k^3$, which is one choice that achieves the fastest convergence rate according to our analysis, gives the best performance across all four data sets. However, in these plots we also see that reasonable performance can be achieved by any of the three strategies above provided that $\alpha$ is chosen carefully. For example, choosing $n = 3$ or choosing $\varepsilon_k = 10^{-6}$ both give reasonable performance. However, these are only empirical observations for these data sets and they may be ineffective for other data sets or if we change the number of iterations, while we have given theoretical justification for the choice $\varepsilon_k = 1/k^3$.

Similar trends are observed for the case of accelerated proximal-gradient methods, though the choice of $\varepsilon_k = 1/k^3$ (which no longer achieves the fastest convergence rate according to our analysis) no longer dominates the other methods in the accelerated setting. For the ***SRBCT*** data set the choice

[1]The datasets are freely available at `http://www.gems-system.org`.

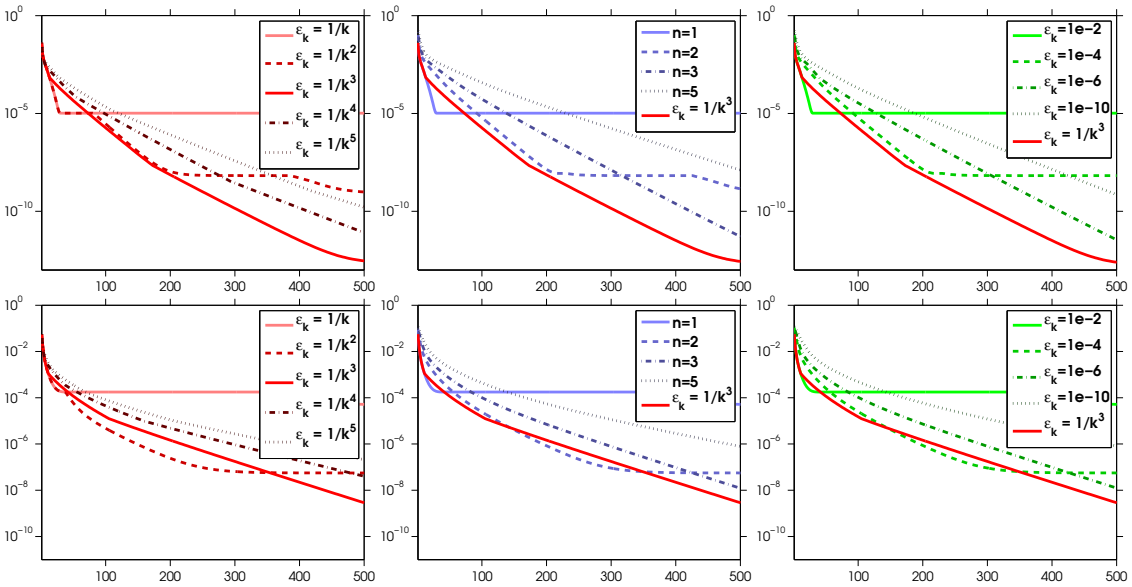


Figure 1: Objective function against number of proximal iterations for the proximal-gradient method with different strategies for terminating the approximate proximity calculation. The top row is for the *9_Tumors* data, the bottom row is for the *Brain_Tumor1* data.

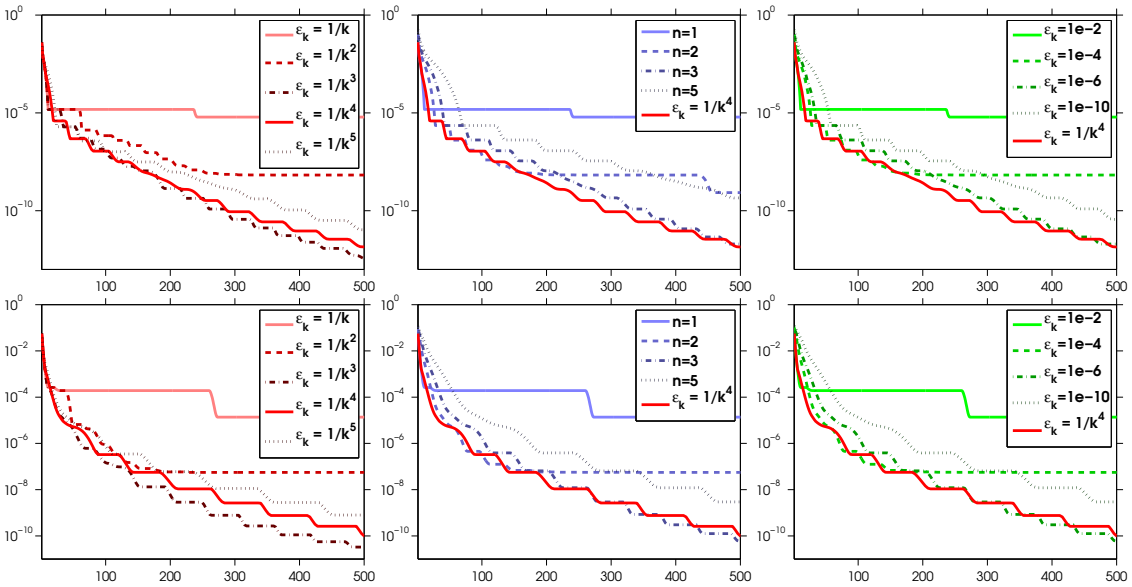


Figure 2: Objective function against number of proximal iterations for the *accelerated* proximal-gradient method with different strategies for terminating the approximate proximity calculation. The top row is for the *9_Tumors* data, the bottom row is for the *Brain_Tumor1* data.

$\varepsilon_k = 1/k^4$, which is a choice that achieves the fastest convergence rate up to a poly-logarithmic factor, yields better performance than $\varepsilon_k = 1/k^3$. Interestingly, the only choice that yields the fastest possible convergence rate ($\varepsilon_k = 1/k^5$) had reasonable performance but did not give the best performance on any data set. This seems to reflect the trade-off between performing *inner* BCD iterations to achieve a small duality gap and performing *outer* gradient iterations to decrease the value of $f$. Also, the constant terms which were not taken into account in the analysis do play an important role here, due to the relatively small number of *outer* iterations performed.

## 6 Discussion

An alternative to inexact proximal methods for solving structured sparsity problems are smoothing methods [35] and alternating direction methods [36]. However, a major disadvantage of both these approaches is that the iterates are not sparse, so they can not take advantage of the sparsity of the problem when running the algorithm. In contrast, the method proposed in this paper has the appealing property that it tends to generate sparse iterates. Further, the accelerated smoothing method only has a convergence rate of $O(1/k)$, and the performance of alternating direction methods is often sensitive to the exact choice of their penalty parameter. On the other hand, while our analysis suggests using a sequence of errors like $O(1/k^\alpha)$ for $\alpha$ large enough, the practical performance of inexact proximal-gradients methods will be sensitive to the exact choice of this sequence.

Although we have illustrated the use of our results in the context of a structured sparsity problem, inexact proximal-gradient methods are also used in other applications such as total-variation [7, 8] and nuclear-norm [9, 10] regularization. This work provides a theoretical justification for using inexact proximal-gradient methods in these and other applications, and suggests some guidelines for practioners that do not want to lose the appealing convergence rates of these methods. Further, although our experiments and much of our discussion focus on errors in the calculation of the proximity operator, our analysis also allows for an error in the calculation of the gradient. This may also be useful in a variety of contexts. For example, errors in the calculation of the gradient arise when fitting undirected graphical models and using an iterative method to approximate the gradient of the log-partition function [37]. Other examples include using a reduced set of training examples within kernel methods [38] or subsampling to solve semidefinite programming problems [39].

In our analysis, we assume that the smoothness constant $L$ is known, but it would be interesting to extend methods for estimating $L$ in the exact case [2] to the case of inexact algorithms. In the context of accelerated methods for strongly convex optimization, our analysis also assumes that $\mu$ is known, and it would be interesting to explore variants that do not make this assumption. We also note that if the basic proximal-gradient method is given knowledge of $\mu$, then our analysis can be modified to obtain a faster linear convergence rate of $(1-\gamma)/(1+\gamma)$ instead of $(1-\gamma)$ for strongly-convex optimization using a step size of $2/(\mu+L)$, see Theorem 2.1.15 of [30]. Finally, we note that there has been recent interest in inexact proximal Newton-like methods [40], and it would be interesting to analyze the effect of errors on the convergence rates of these methods.

**Acknowledgements** Mark Schmidt, Nicolas Le Roux, and Francis Bach are supported by the European Research Council (SIERRA-ERC-239993).

## References

[1] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.

[2] Y. Nesterov. Gradient methods for minimizing composite objective function. *CORE Discussion Papers*, (2007/76), 2007.

[3] R. Tibshirani. Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society: Series B*, 58(1):267–288, 1996.

[4] S.S. Chen, D.L. Donoho, and M.A. Saunders. Atomic decomposition by basis pursuit. *SIAM Journal on Scientific Computing*, 20(1):33–61, 1998.

[5] S.J. Wright, R.D. Nowak, and M.A.T. Figueiredo. Sparse reconstruction by separable approximation. *IEEE Transactions on Signal Processing*, 57(7):2479–2493, 2009.

[6] F. Bach, R. Jenatton, J. Mairal, and G. Obozinski. Convex optimization with sparsity-inducing norms. In S. Sra, S. Nowozin, and S.J. Wright, editors, *Optimization for Machine Learning*. MIT Press, 2011.

[7] J. Fadili and G. Peyré. Total variation projection with first order schemes. *IEEE Transactions on Image Processing*, 20(3):657–669, 2011.

[8] X. Chen, S. Kim, Q. Lin, J.G. Carbonell, and E.P. Xing. Graph-structured multi-task regression and an efficient optimization method for general fused Lasso. *arXiv:1005.3579v1*, 2010.

[9] J.-F. Cai, E.J. Candès, and Z. Shen. A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization*, 20(4), 2010.

[10] S. Ma, D. Goldfarb, and L. Chen. Fixed point and Bregman iterative methods for matrix rank minimization. *Mathematical Programming*, 128(1):321–353, 2011.

[11] L. Jacob, G. Obozinski, and J.-P. Vert. Group Lasso with overlap and graph Lasso. *ICML*, 2009.

[12] R. Jenatton, J. Mairal, G. Obozinski, and F. Bach. Proximal methods for sparse hierarchical dictionary learning. *JMLR*, 12:2297–2334, 2011.

[13] A. Barbero and S. Sra. Fast Newton-type methods for total variation regularization. *ICML*, 2011.

[14] J. Liu and J. Ye. Fast overlapping group Lasso. *arXiv:1009.0306v1*, 2010.

[15] M. Schmidt and K. Murphy. Convex structure learning in log-linear models: Beyond pairwise potentials. *AISTATS*, 2010.

[16] M. Patriksson. *A unified framework of descent algorithms for nonlinear programs and variational inequalities*. PhD thesis, Department of Mathematics, Linköping University, Sweden, 1995.

[17] P.L. Combettes. Solving monotone inclusions via compositions of nonexpansive averaged operators. *Optimization*, 53(5-6):475–504, 2004.

[18] J. Duchi and Y. Singer. Efficient online and batch learning using forward backward splitting. *JMLR*, 10:2873–2898, 2009.

[19] J. Langford, L. Li, and T. Zhang. Sparse online learning via truncated gradient. *JMLR*, 10:777–801, 2009.

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

[21] M. Baes. Estimate sequence methods: extensions and approximations. IFOR internal report, ETH Zurich, 2009.

[22] O. Devolder, F. Glineur, and Y. Nesterov. First-order methods of smooth convex optimization with inexact oracle. *CORE Discussion Papers*, (2011/02), 2011.

[23] A. Nedic and D. Bertsekas. Convergence rate of incremental subgradient algorithms. *Stochastic Optimization: Algorithms and Applications*, pages 263–304, 2000.

[24] Z.-Q. Luo and P. Tseng. Error bounds and convergence analysis of feasible descent methods: A general approach. *Annals of Operations Research*, 46-47(1):157–178, 1993.

[25] M.P. Friedlander and M. Schmidt. Hybrid deterministic-stochastic methods for data fitting. *arXiv:1104.2373*, 2011.

[26] R.T. Rockafellar. Monotone operators and the proximal point algorithm. *SIAM Journal on Control and Optimization*, 14(5):877–898, 1976.

[27] O. Güler. New proximal point algorithms for convex minimization. *SIAM Journal on Optimization*, 2(4):649–664, 1992.

[28] S. Villa, S. Salzo, L. Baldassarre, and A. Verri. Accelerated and inexact forward-backward algorithms. *Optimization Online*, 2011.

[29] M. Schmidt, N. Le Roux, and F. Bach. Convergence rates of inexact proximal-gradient methods for convex optimization. *arXiv:1109.2415v2*, 2011.

[30] Y. Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*. Springer, 2004.

[31] D.P. Bertsekas. *Convex optimization theory*. Athena Scientific, 2009.

[32] P. Tseng. On accelerated proximal gradient methods for convex-concave optimization, 2008.

[33] J. Mairal, R. Jenatton, G. Obozinski, and F. Bach. Convex and network flow optimization for structured sparsity. *JMLR*, 12:2681–2720, 2011.

[34] H.H. Bauschke and P.L. Combettes. A Dykstra-like algorithm for two monotone operators. *Pacific Journal of Optimization*, 4(3):383–391, 2008.

[35] Y. Nesterov. Smooth minimization of non-smooth functions. *Math. Prog.*, 103(1):127–152, 2005.

[36] P.L. Combettes and J.-C. Pesquet. Proximal splitting methods in signal processing. In H.H. Bauschke, R.S. Burachik, P.L. Combettes, V. Elser, D.R. Luke, and H. Wolkowicz, editors, *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, pages 185–212. Springer, 2011.

[37] M.J. Wainwright, T.S. Jaakkola, and A.S. Willsky. Tree-reweighted belief propagation algorithms and approximate ML estimation by pseudo-moment matching. *AISTATS*, 2003.

[38] J. Kivinen, A.J. Smola, and R.C. Williamson. Online learning with kernels. *IEEE Transactions on Signal Processing*, 52(8):2165–2176, 2004.

[39] A. d'Aspremont. Subsampling algorithms for semidefinite programming. *arXiv:0803.1990v5*, 2009.

[40] M. Schmidt, D. Kim, and S. Sra. Projected Newton-type methods in machine learning. In S. Sra, S. Nowozin, and S. Wright, editors, *Optimization for Machine Learning*. MIT Press, 2011.