An Inexact Successive Quadratic Approximation Method for Convex L-1 Regularized Optimization

Richard H. Byrd* Jorge Nocedal † Figen Oztoprak‡

September 7, 2013

Abstract

We study a Newton-like method for the minimization of an objective function ϕ that is the sum of a smooth convex function and an ℓ_1 regularization term. This method, which is sometimes referred to in the literature as a proximal Newton method, computes a step by minimizing a piecewise quadratic model q_k of the objective function ϕ. In order to make this approach efficient in practice, it is imperative to perform this inner minimization inexactly. In this paper, we give inexactness conditions that guarantee global convergence and that can be used to control the local rate of convergence of the iteration. Our inexactness conditions are based on a semi-smooth function that represents a (continuous) measure of the optimality conditions of the problem, and that embodies the soft-thresholding iteration. We give careful consideration to the algorithm employed for the inner minimization, and report numerical results on two test sets originating in machine learning.

1 Introduction

In this paper, we study an inexact Newton-like method for solving optimization problems of the form

$$\min_{x\in\mathbb{R}^n} \ \phi(x) = f(x) + \mu\|x\|_1, \tag{1.1}$$

where f is a smooth convex function and $\mu > 0$ is a (fixed) regularization parameter. The method constructs, at every iteration, a piecewise quadratic model of ϕ and minimizes this model *inexactly* to obtain a new estimate of the solution.

The piecewise quadratic model is defined, at an iterate x_k, as

$$q_k(x) = f(x_k) + g(x_k)^T(x - x_k) + \tfrac{1}{2}(x - x_k)^T H_k(x - x_k) + \mu\|x\|_1, \tag{1.2}$$

*Department of Computer Science, University of Colorado, Boulder, CO, USA. This author was supported by National Science Foundation grant CMMI 0728190 and Department of Energy grant DE-SC0001774.

†Department of Industrial Engineering and Management Sciences, Northwestern University, Evanston, IL, USA. This author was supported by National Science Foundation grant DMS-0810213, and by Department of Energy grant DE-FG02-87ER25047.

‡Istanbul Technical University. This author was supported by Department of Energy grant DE-SC0001774, and by a grant from Tubitak.

where $g(x_k) \stackrel{\text{def}}{=} \nabla f(x_k)$ and H_k denotes the Hessian $\nabla^2 f(x_k)$ or a quasi-Newton approximation to it. After computing an approximate solution $\hat{x}$ of this model, the algorithm performs a backtracking line search along the direction $d_k = \hat{x} - x_k$ to ensure decrease in the objective ϕ.

We refer to this method as the *successive quadratic approximation method* in analogy to the successive quadratic programming method for nonlinear programming. This method is also known in the literature as a "proximal Newton method" [20, 24], but we prefer not to use the term "proximal" in this context since the quadratic term in (1.2) is better interpreted as a second-order model rather than as a term that simply restricts the size of the step. The paper covers both the cases when the quadratic model q_k is constructed with an exact Hessian or a quasi-Newton approximation.

The two crucial ingredients in the inexact successive quadratic approximation method are the algorithm used for the minimization of the model q_k, and the criterion that controls the degree of inexactness in this minimization. In the first part of the paper, we propose an inexactness criterion for the minimization of q_k and prove that it guarantees global convergence of the iterates, and that it can be used to control the local rate of convergence. This criterion is based on the optimality conditions for the minimization of (1.2), expressed in the form of a semi-smooth function that is derived from the soft-thresholding operator.

The second part of the paper is devoted to the practical implementation of the method. Here, the choice of algorithm for the inner minimization of the model q_k is vital, and we consider two options: FISTA [4], which is a first-order method, and an orthant-based method [1, 7, 8]. The latter is a second-order method where each iteration consists of an orthant-face identification phase, followed by the minimization of a smooth model restricted to that orthant. The subspace minimization can be performed by computing a quasi-Newton step or a Newton-CG step (we explore both options). A projected bactracking line search is then applied; see section 5.3.

Some recent work on successive quadratic approximation methods for problem (1.1) include: Hsie et al. [12], where (1.2) is solved using a coordinate descent method, and which focuses on the inverse covariance selection method; [25] which also employs coordinate descent but uses a different working set identification than [12], and makes use of a quasi-Newton model; and Olsen et al. [18], where the inner solver is FISTA. None of these papers address convergence for inexact solutions of the subproblem. Recently Lee, Sun and Saunders [13] presented an inexact proximal Newton method that, at first glance, appears to be very close to the method presented here. Their inexactness criterion is, however, different from ours and suffers from a number of drawbacks, as discussed in section 2.

Inexact methods for solving generalized equations have been studied by Patricksson [21], and more recently by Dontchev and Rockafellar [10]. Special cases of the general methods described in those papers result in inexact sequential quadratic approximation algorithms. Patricksson [21] presents convergence analyses based on two conditions for controlling inexactness. The first is based on running the subproblem solver for a limited number of steps. The second rule requires that the residual norm be sufficiently small, but it does not cover the inexactness conditions presented in this paper (since the residual is computed differently and their inexactness measure is is different from ours). The rule

suggested in Dontchev and Rockafellar [10] is very general, but it too does not cover the condition presented in this paper. Our rule, and those presented in [10, 13], is inspired by the classical inexactness condition proposed by Dembo et al. [9], and reduces to it for the smooth unconstrained minimization case (i.e. when $\mu = 0$).

Another line of research that is relevant to this paper is the global and rate of convergence analysis for inexact proximal-gradient algorithms, which can be seen as special cases of sequential quadratic approximation without acceleration [15, 23, 26]. The inexactness conditions applied in those papers require that the subproblem objective function value be ϵ-close to the optimal subproblem objective [15, 26], or that the approximate solution be exact with respect to an ϵ-perturbed subdifferential [23], for a decreasing sequence $\{\epsilon\}$.

Our interest in the successive quadratic approximation method is motivated by the fact that it has not received sufficient attention from a practical perspective, where inexact solutions to the inner problem (1.2) are imperative. Although a number of studies have been devoted to the formulation and analysis of proximal Newton methods for convex composite optimization problems, as mentioned above, the viability of the approach in practice has not been fully explored.

This paper is organized in 5 sections. In section 2 we outline the algorithm, including the inexactness criteria that govern the solution of the subproblem (1.2). In sections 3 and 4, we analyze the global and local convergence properties of the algorithm. Numerical experiments are reported in section 5. The paper concludes in section 6 with a summary of our findings, and a list of questions to explore.

Notation. In the remainder, we let $g(x_k) = \nabla f(x_k)$, and let $\|\cdot\|$ denote any vector norm. We sometimes abbreviate successive quadratic approximation method as "SQA method", and note that this algorithm is often referred to in the literature as the "proximal Newton method".

2 The Algorithm

Given an iterate x_k, an iteration of the algorithm begins by forming the model (1.2), where $\mu > 0$ is a given scalar and $H_k \succ 0$ is an approximation to the Hessian $\nabla^2 f(x_k)$. Next, the algorithm computes an *approximate* minimizer $\hat{x}$ of the subproblem

$$\min_{x \in \mathbb{R}^n} \; q_k(x) = f(x_k) + g(x_k)^T(x - x_k) + \tfrac{1}{2}(x - x_k)^T H_k(x - x_k) + \mu\|x\|_1. \qquad (2.1)$$

The point $\hat{x}$ defines the search direction $d_k = \hat{x} - x_k$. The algorithm then performs a backtracking line search along the direction d_k that ensures sufficient decrease in the objective ϕ. The minimization of (2.1) should be performed by a method that exploits the structure of this problem.

In order to compute an adequate approximate solution to (1.2), we need some measure of closeness to optimality. In the case of smooth unconstrained optimization, (i.e. (1.1) with $\mu = 0$), the norm of the gradient is a standard measure of optimality, and it is common [9] to require the approximate solution $\hat{x}$ to satisfy the condition

$$\|g(x_k) + H_k(\hat{x} - x_k)\| \leq \eta_k \|g(x_k)\|, \qquad 0 < \eta_k < 1. \qquad (2.2)$$

The term on the left side of (2.2) is a measure of optimality for the model $q_k(x)$, in the unconstrained case.

For problem (1.1), the length of the iterative soft-thresholding (ISTA) step is a natural measure of optimality. The ISTA iteration is given by

$$x_{\text{ista}} = \arg\min_x \; g(x_k)^T(x - x_k) + \frac{1}{2\tau}\|x - x_k\|^2 + \mu\|x\|_1, \tag{2.3}$$

where $\tau > 0$ is a fixed parameter. It is easy to verify that $\|x_{\text{ista}} - x_k\|$ is zero if and only if x_k is a solution of problem (1.1). We need to express $\|x_{\text{ista}} - x_k\|$ in a way that is convenient for our analysis, and for this purpose we note [16] that some algebraic manipulations show that $\|x_{\text{ista}} - x_k\| = \tau\|F(x_k)\|$, where

$$F(x) = g(x) - P_{[-\mu,\mu]}(g(x) - x/\tau). \tag{2.4}$$

Here $P(x)_{[-\mu,\mu]}$ denotes the component-wise projection of x onto the interval $[-\mu, \mu]$, and τ is a positive scalar.

One can directly verify that (2.4) is a valid optimality measure by noting that $F(x) = 0$ is equivalent to the standard necessary optimality condition for (1.1):

$$\begin{aligned} g_i(x^*) + \mu &= 0 \quad \text{for } i \text{ s.t. } x_i^* > 0, \\ g_i(x^*) - \mu &= 0 \quad \text{for } i \text{ s.t. } x_i^* < 0, \\ -\mu \le g_i(x^*) &\le \mu \quad \text{for } i \text{ s.t. } x_i^* = 0. \end{aligned}$$

For the objective q_k of (2.1), this function takes the form

$$F_q(x_k; x) = g(x_k) + H_k(x - x_k) - P_{[-\mu,\mu]}(g(x_k) + H_k(x - x_k) - x/\tau). \tag{2.5}$$

Using the measures (2.4) and (2.5) in a manner similar to (2.2), leads to the condition $\|F_q(x_k; \hat{x})\| \le \eta_k\|F(x_k)\|$. However, depending on the method used to approximately solve (2.1), this does not guarantee that $\hat{x} - x_k$ is a descent direction for ϕ. To achieve this, we impose the additional condition that the quadratic model is decreased at $\hat{x}$.

Inexactness Conditions. A point $\hat{x}$ is considered an acceptable approximate solution of subproblem (2.1) if

$$\|F_q(x_k; \hat{x})\| \le \eta_k\|F_q(x_k; x_k)\| \quad \text{and} \quad q_k(\hat{x}) < q_k(x_k), \tag{2.6}$$

for some parameter $0 \le \eta_k < 1$, where $\|\cdot\|$ is any norm. (Note that $F_q(x_k; x_k) = F(x_k)$, so that the first condition can also be written as $\|F_q(x_k; \hat{x})\| \le \eta_k\|F(x_k)\|$.)

The method is summarized in Algorithm 2.1.

Algorithm 2.1: Inexact Successive Quadratic Approximation (SQA) Method for Problem (1.1)

Choose an initial iterate x_0.
Select constants $\theta \in (0, 1/2)$ and $0 < \tau < 1$ (which is used in definitions (2.4), (2.5)).

for $k = 0, \cdots$, until the optimality conditions of (1.1) are satisfied:

1. Compute (or update) the model Hessian H_k and form the piecewise quadratic model (1.2);

2. Compute an inexact solution $\hat{x}$ of (1.2) satisfying conditions (2.6).

3. Perform a backtracking line search along the direction $d = \hat{x} - x_k$: starting with $\alpha = 1$, find $\alpha \in (0, 1]$ such that

$$\phi(x_k) - \phi(x_k + \alpha d) \geq \theta(\ell_k(x_k) - \ell_k(x_k + \alpha d)), \tag{2.7}$$

   where ℓ is the following piecewise linear model of ϕ at x_k:

$$\ell_k(x) = f(x_k) + g(x_k)^T(x - x_k) + \mu\|x\|_1. \tag{2.8}$$

4. Set $x_{k+1} = x_k + \alpha d$.

end(for)

For now, we simply assume that the sequence $\{\eta_k\}$ in (2.6) satisfies $\eta_k \in [0, 1)$, but in section 4 we show that by choosing $\{\eta_k\}$ and the parameter τ appropriately, the algorithm achieves a fast rate of convergence. One may wonder whether the backtracking line search of Step 3 might hinder sparsity of the iterates. Our numerical experience indicates that this is not the case because, in our tests, Algorithm 2.1 almost always accepts the unit steplength ($\alpha = 1$).

It is worth pointing out that Lee et al. [13] recently proposed and analyzed an inexactness criterion that is similar to the first inequality of (2.6). The main difference is that they use the subgradient of q_k on the left side of the inequality, and both norms are scaled by H_k^{-1}. They claim similar convergence results to ours, but a worrying consequence of the lack of continuity of the subgradient of q_k is that their inexactness condition can fail for vectors x arbitrarily close to the exact minimizer of q_k. As a result, their criterion is not an appropriate termination test for the inner iteration. (In addition, their use of the scaling H_k^{-1} precludes setting $H_k = \nabla^2 f(x_k)$, except for small or highly structured problems.)

3 Global Convergence

In this section, we show that Algorithm 2.1 is globally convergent under certain assumptions on the function f and the (approximate) Hessians H_k. Specifically, we assume that f is a differentiable function with Lipschitz continuous gradient, i.e., there is a constant $M > 0$ such that

$$\|g(x) - g(y)\| \leq M\|x - y\|, \tag{3.1}$$

for all x, y. We denote by $\lambda_{\min}(H_k)$ and $\lambda_{\max}(H_k)$ the smallest and largest eigenvalues of H_k, respectively.

Theorem 3.1 *Suppose that f is a smooth function that is bounded below and that satisfies (3.1). Let $\{x_k\}$ be the sequence of iterates generated by Algorithm 2.1, and suppose that there exist constants $0 < \lambda \leq \Lambda$ such that the sequence $\{H_k\}$ satisfies*

$$\lambda_{\min}(H_k) \geq \lambda > 0 \quad \textit{and} \quad \lambda_{\max}(H_k) \leq \Lambda,$$

for all k. Then

$$\lim_{k\to\infty} F(x_k) = 0. \tag{3.2}$$

Proof. We first show that if $\hat{x}$ is an approximate solution of (1.2) that satisfies the inexactness conditions (2.6), then there is a constant $\gamma > 0$ (independent of k) such that for all $k \in \{0, 1, \cdots\}$

$$\ell_k(x_k) - \ell_k(\hat{x}) \geq \gamma\|F(x_k)\|^2, \tag{3.3}$$

where ℓ_k and F are defined in (2.8) and (2.4). To see this, note that by (2.6)

$$0 > q_k(\hat{x}) - q_k(x_k) = \ell_k(\hat{x}) - \ell_k(x_k) + \tfrac{1}{2}(\hat{x} - x_k)^T H_k(\hat{x} - x_k),$$

and therefore

$$\ell_k(x_k) - \ell_k(\hat{x}) > \tfrac{1}{2}(\hat{x} - x_k)^T H_k(\hat{x} - x_k) \geq \tfrac{1}{2}\lambda\|\hat{x} - x_k\|^2. \tag{3.4}$$

Next, since $F(x_k) = F_q(x_k; x_k)$, and using (2.6) and the contraction property of the projection, we have that

$$\begin{aligned}
(1 - \eta_k)\|F(x_k)\| &= (1 - \eta_k)\|F_q(x_k; x_k)\| \\
&\leq \|F_q(x_k; x_k)\| - \|F_q(x_k, \hat{x})\| \\
&\leq \|F_q(x_k; \hat{x}) - F_q(x_k; x_k)\| \\
&= \|H_k(\hat{x} - x_k) - P_{[-\mu,\mu]}(g(x_k) + H_k(\hat{x} - x_k) - \tfrac{1}{\tau}\hat{x}) + P_{[-\mu,\mu]}(g(x_k) - \tfrac{1}{\tau}x_k)\| \\
&\leq \|H_k(\hat{x} - x_k)\| + \|\tfrac{1}{\tau}(\hat{x} - x_k) - H_k(\hat{x} - x_k)\| \\
&\leq \tfrac{1}{\tau}\|\hat{x} - x_k\| + 2\|H_k\|\|\hat{x} - x_k\| \\
&= \left(\tfrac{1}{\tau} + 2\|H_k\|\right)\|\hat{x} - x_k\| \\
&\leq \left(\tfrac{1}{\tau} + 2\Lambda\right)\|\hat{x} - x_k\|.
\end{aligned}$$

Combining this expression with (3.4), we obtain (3.3) for

$$\gamma = \frac{\lambda}{2}\left(\frac{1-\eta}{\frac{1}{\tau}+2\Lambda}\right)^2.$$

Note that $\gamma > 0$ as $\tau, \lambda, \Lambda > 0$ and $\eta \in [0, 1)$.

Let us define the search direction as $d = \hat{x} - x_k$. We now show that by performing a line search along d we can ensure that the algorithm provides sufficient decrease in the objective function ϕ, and this will allow us to establish the limit (3.2).

Since $g(x)$ satisfies the Lipschitz condition (3.1), we have

$$f(x_k + \alpha d) + \mu\|x_k + \alpha d\|_1 \leq f(x_k) + \alpha g(x_k)^T d + \frac{M}{2}\alpha^2\|d\|^2 + \mu\|x_k + \alpha d\|_1,$$

and thus

$$f(x_k)+\mu\|x_k\|_1-f(x_k+\alpha d)-\mu\|x_k+\alpha d\|_1 \geq -\alpha g(x_k)^T d-\frac{M}{2}\alpha^2\|d\|^2-\mu\|x_k+\alpha d\|_1+\mu\|x_k\|_1.$$

Recalling the definition of ℓ_k, we have

$$\phi(x_k) - \phi(x_k + \alpha d) \geq \ell_k(x_k) - \ell_k(x_k + \alpha d) - \frac{M}{2}\alpha^2\|d\|^2.$$

By convexity of the ℓ_1-norm, we have that

$$\ell(x_k) - \ell(x_k + \alpha d) \geq \alpha(\ell(x_k) - \ell(x_k + d)).$$

Combining this inequality with (3.4), and recalling that $x + d = \hat{x}$, we obtain for $\theta \in (0, 1)$,

$$\begin{aligned}
\phi(x_k) - \phi(x_k + \alpha d) - \theta(\ell(x_k) - \ell(x_k + \alpha d)) &\geq (1-\theta)(\ell(x_k) - \ell(x_k + \alpha d)) - \frac{M}{2}\alpha^2\|d\|^2 \\
&\geq (1-\theta)\alpha(\ell(x_k) - \ell(x_k + d)) - \frac{M}{2}\alpha^2\|d\|^2 \\
&\geq (1-\theta)\alpha\frac{\lambda}{2}\|d\|^2 - \frac{M}{2}\alpha^2\|d\|^2 \\
&= \tfrac{1}{2}\alpha\|d\|^2\left((1-\theta)\lambda - M\alpha\right) \\
&\geq 0, \qquad (3.5)
\end{aligned}$$

provided $((1-\theta)\lambda - M\alpha) \geq 0$. Therefore, the sufficient decrease condition (2.7) is satisfied for any steplength α satisfying

$$0 \leq \alpha \leq (1-\theta)\frac{\lambda}{M},$$

and if the backtracking line search cuts the steplength in half (say) after each trial, we have that the steplength chosen by the line search satisfies

$$\alpha \geq (1-\theta)\frac{\lambda}{2M}.$$

Thus, from (3.5) and (3.3) we obtain

$$\phi(x_k) - \phi(x_k + \alpha d) \geq \theta(1-\theta)\frac{\lambda}{2M}\gamma\|F(x_k)\|^2.$$

Since f is assumed to be bounded below, so is the objective function ϕ, and given that the decrease in ϕ is proportional to $\|F(x_k)\|$ we obtain the limit (3.2). □

We note that to establish this convergence result it was not necessary to assume convexity of f.

4 Local Convergence

To analyze the local convergence rate of the successive quadratic approximation method, we use the theory developed in Chapter 7 of Facchinei and Pang [11]. To do this, we first show that, if x^* is a nonsingular minimizer of ϕ, then the functions $F_q(x;\cdot) : \mathbb{R}^n \to \mathbb{R}^n$ are a family of uniformly Lipschitzian nonsingular homeomorphisms for all x in a neighborhood of x^*.

Lemma 4.1 *If H is a symmetric positive definite matrix with smallest eigenvalue $\lambda > 0$, then the function of y given by*

$$F_q(x;y) = g(x) + H(y-x) - P_{[-\mu,\mu]}(g(x) + H(y-x) - y/\tau),$$

is strongly monotone if $\tau < 1/\|H\|$. Specifically, for any vectors $y, z \in \mathbb{R}^n$,

$$(z-y)^T(F_q(x;z) - F_q(x;y)) \geq \tfrac{1}{2}\lambda\|z-y\|^2. \tag{4.1}$$

Proof. It is straightforward to show that for any scalars $a \neq b$ and interval $[-\mu,\mu]$,

$$0 \leq \frac{P_{[-\mu,\mu]}(a) - P_{[-\mu,\mu]}(b)}{a-b} \leq 1. \tag{4.2}$$

Therefore for any vectors y and z, and for any index $i \in \{1,\cdots,n\}$ we have

$$\begin{aligned} F_q(x;z)_i - F_q(x;y)_i &= H(z-y)_i - P_{[-\mu,\mu]}(g_i(x) + H(z-x)_i - z_i/\tau)] \\ &\quad + P_{[-\mu,\mu]}(g_i(x) + H(y-x)_i - y_i/\tau)] \\ &= H(z-y)_i - \bar{d}_i[H(z-y)_i - (z_i - y_i)/\tau], \end{aligned}$$

where $\bar{d}_i \in [0,1]$ is a scalar implied by (4.2). This implies that

$$F_q(x;z) - F_q(x;y) = H(z-y) + D(\tfrac{1}{\tau}I - H)(z-y),$$

where $D = \mathrm{diag}(\bar{d}_i)$. Hence

$$(z-y)^T(F_q(x;z) - F_q(x;y)) = (z-y)^T H(z-y) + (z-y)^T D(\tfrac{1}{\tau}I - H)(z-y). \tag{4.3}$$

Since the right hand side is a quadratic form, we symmetrize the matrix, and if we let $w = z - y$, the right side is

$$w^T[H + \tau^{-1}D - \tfrac{1}{2}(DH + HD)]w. \tag{4.4}$$

To show that the symmetric matrix inside the square brackets is positive definite, we note that since $(\tau H - D)^T(\tau H - D) = \tau^2 H^2 - \tau(HD + DH) + D^2$ is positive semi-definite, we have that

$$w^T(HD + DH)w \leq w^T(\tau H^2 + \tau^{-1}D^2)w.$$

Substituting this into (4.4) yields

$$\begin{aligned} w^T\left[H + \tau^{-1}D - \tfrac{1}{2}(DH + HD)\right]w &\geq w^T\left[H - \frac{\tau}{2}H^2 + \frac{D - D^2/2}{\tau}\right]w \\ &\geq w^T\left[H - \frac{\tau}{2}H^2\right]w, \end{aligned}$$

since $D - D^2/2$ is positive semi-definite given that the elements of the diagonal matrix D are in $[0, 1]$. If λ_i is an eigenvalue of H, the corresponding eigenvalue of the matrix $H - \frac{\tau}{2}H^2$ is $\lambda_i - \tau\lambda_i^2/2 \geq \lambda_i/2$ since our assumption on τ implies $1 > \tau\|H\| \geq \tau\lambda_i$. Therefore, we have from (4.3) that

$$(z - y)^T(F_q(x; z) - F_q(x; y)) \geq \tfrac{1}{2}\lambda\|z - y\|^2.$$

□

Inequality (4.1) establishes that $F_q(x; \cdot)$ is strongly monotone. Next we show that, when H is defined as the Hessian of f, the functions $F_q(x; \cdot)$ are homeomorphisms and that they represent an accurate approximation to the function F defined in (2.4).

Theorem 4.2 *If $\nabla^2 f(x^*)$ is positive definite and $\tau < 1/\|\nabla^2 f(x^*)\|$, then there is a neighborhood $\mathcal{N}$ of x^* such that for all $x \in \mathcal{N}$ the functions of y given by*

$$F_q(x; y) = g(x) + \nabla^2 f(x)(y - x) - P_{[-\mu,\mu]}(g(x) + \nabla^2 f(x)(y - x) - y/\tau) \tag{4.5}$$

are a family of homeomorphisms from $\mathbb{R}^n$ to $\mathbb{R}^n$, whose inverses $F_q^{-1}(x; \cdot)$ are uniformly Lipschitz continuous. In addition, if $\nabla^2 f(x)$ is Lipschitz continuous, then there exists a constant $\beta > 0$ such that

$$\|F(y) - F_q(x; y)\| \leq \beta\|x - y\|^2 \tag{4.6}$$

for any $x \in \mathcal{N}$.

Proof. Since $\nabla^2 f(x)$ is continuous, there is a neighborhood $\mathcal{N}$ of x^* and a positive constant λ such that $\lambda_{min}(\nabla^2 f(x)) \geq \lambda > 0$ and $\tau\|\nabla^2 f(x)\| < 1$, for all $x \in \mathcal{N}$. It follows from Lemma 4.1 that for any such x, the function $F_q(x; y)$ given by (4.5) is strongly (or uniformly) monotone with constant greater than $\lambda/2$. We now invoke the Uniform Monotonicity Theorem (see e.g. Theorem 6.4.4 in [19]), which states that if a function $F : \mathbb{R}^n \to \mathbb{R}^n$ is

continuous and uniformly monotone, then F is a homeomorphism of $\mathbb{R}^n$ onto itself. We therefore conclude that $F_q(x;y)$ is a homeomorphism.

In addition, we have from (4.1) and the Cauchy-Schwartz inequality that

$$\|z-y\|\|F_q(x;z)-F_q(x;y)\| \geq (z-y)^T(F_q(x;z)-F_q(x;y)) \geq \tfrac{1}{2}\lambda\|z-y\|^2,$$

which implies Lipschitz continuity of $F_q^{-1}(x;\cdot)$ with constant $2/\lambda$. To establish (4.6), note that

$$\begin{aligned} F(y)-F_q(x;y) = g(y) - P_{[-\mu,\mu]}(g(y)-y/\tau) - \big(g(x)+\nabla^2 f(x)(y-x)\big) \\ + P_{[-\mu,\mu]}(g(x)+\nabla^2 f(x)(y-x)-y/\tau), \end{aligned}$$

and thus

$$\begin{aligned} \|F(y)-F_q(x;y)\| &\leq \|g(y)-(g(x)+\nabla^2 f(x)(y-x))\| \\ &\quad +\|P_{[-\mu,\mu]}(g(y)-y/\tau)-P_{[-\mu,\mu]}(g(x)+\nabla^2 f(x)(y-x)-y/\tau)\| \\ &\leq 2\|g(y)-(g(x)+\nabla^2 f(x)(y-x))\| \\ &= O(\|y-x\|^2), \end{aligned}$$

by the non-expansiveness of a projection onto a convex set and Taylor's theorem. □

Theorem 4.2 shows that $F_q(x;y)$ defines a strong nonsingular Newton approximation in the sense of Definition 7.2.2 of Pang and Facchinei [11]. This implies quadratic convergence for the (exact) successive quadratic approximation (SQA) method.

Theorem 4.3 *If $\nabla^2 f(x)$ is Lipschitz continuous and positive definite at x^*, and $\tau < 1/\|\nabla^2 f(x^*)\|$, then there is a neighborhood of x^* such that, if x_0 lies in that neighborhood, the iteration that defines x_{k+1} as the unique solution to*

$$F_q(x_k;x_{k+1}) = 0$$

converges quadratically to x^*.

Proof. By Theorem 4.2, $F_q(x_k;y)$ satisfies the definition of a nonsingular strong Newton approximation of F at x^*, given by Facchinei and Pang ([11], 7.2.2) and thus by Theorem 7.2.5 of that book the local convergence is quadratic. □

Now we consider the inexact SQA algorithm that, at each step, computes a point y satisfying

$$F_q(x_k;y) = r_k, \tag{4.7}$$

where r_k is a vector such that $\|r_k\| \leq \eta_k\|F(x_k)\|$ with $\eta_k < 1$; see (2.6). We obtain the following result for a method that sets $x_{k+1} = y$.

Theorem 4.4 *Suppose that $\nabla^2 f(x)$ is Lipschitz continuous and positive definite at x^*, $\tau < 1/\|\nabla^2 f(x^*)\|$, and that x_{k+1} is computed by solving*

$$F_q(x_k;x_{k+1}) = r_k,$$

where $\|r_k\| \leq \eta_k \|F(x_k)\|$. *Then, there is a neighborhood* $\mathcal{N}$ *of* x^* *and a value* $\bar{\eta} > 0$ *such that if* $\eta_k \leq \bar{\eta}$ *for all* k *and if* $x_0 \in \mathcal{N}$ *then the sequence* $\{x_k\}$ *converges Q-linearly to* x^*. *In addition if* $\eta_k \to 0$, *then the convergence rate of* $\{x_k\}$ *is Q-superlinear. Finally, if for some* $\tilde{\eta}$, $\eta_k \leq \tilde{\eta}\|F(x_k)\|$ *then the convergence rate is Q-quadratic.*

Proof. By Theorem 4.2, the iteration described in the statement of the theorem satisfies all the conditions of Theorem 7.2.8 of [11]. The results then follow immediately from that theorem. □

We have shown above the the inexact successive quadratic approximation (SQA) method with $\alpha_k = 1$ yields a fast rate of convergence. We now show that this inexact SQA algorithm will select the steplength $\alpha_k = 1$ in a neighborhood of the solution. In order to do so, we strengthen the inexactness conditions (2.6) slightly so that they read

$$\|F_q(x_k; \hat{x})\| \leq \eta_k \|F_q(x_k; x_k)\| \quad \text{and} \quad q_k(\hat{x}) - q_k(x_k) \leq \zeta(\ell_k(\hat{x}) - \ell_k(x_k)), \tag{4.8}$$

where $\eta_k < 1$, $\zeta \in (\theta, 1/2)$ and θ is the input parameter of Algorithm 2.1 used in (2.7). Thus, instead of simple decrease, we now impose sufficient decrease in q_k.

Lemma 4.5 *If* H_k *is positive definite, the inexactness condition* (4.8) *is satisfied by any sufficiently accurate solution to* (2.1).

Proof. If we denote by $\bar{y}$ the (exact) minimizer of q_k, we claim that

$$q_k(x_k) - q_k(\bar{y}) \geq \tfrac{1}{2}(\ell_k(x_k) - \ell_k(\bar{y})). \tag{4.9}$$

To see this, note that $q_k(y) = \ell_k(y) + \frac{1}{2}(y - x_k)^T H_k(y - x_k)$, and since ℓ_k and q_k are convex and $\bar{y}$ minimizes q_k, there exists a vector $v \in \partial \ell_k(\bar{y})$ such that $v + H_k(\bar{y} - x_k) = 0$. By convexity

$$\ell_k(x_k) \geq \ell_k(\bar{y}) + v^T(x_k - \bar{y}) = \ell_k(\bar{y}) + (\bar{y} - x_k)^T H_k(\bar{y} - x_k). \tag{4.10}$$

Therefore,

$$q_k(x_k) - q_k(\bar{y}) = \ell_k(x_k) - \ell_k(\bar{y}) - \tfrac{1}{2}(\bar{y} - x_k)^T H_k(\bar{y} - x_k) \geq \tfrac{1}{2}(\ell_k(x_k) - \ell_k(\bar{y})),$$

which proves (4.9).

Now consider the continuous function $q_k(x) - q_k(x_k) - \zeta(\ell_k(x) - \ell_k(x_k))$. By (4.9) its value at $x = \bar{y}$ is

$$q_k(\bar{y}) - q_k(x_k) - \zeta(\ell_k(\bar{y}) - \ell_k(x_k)) \leq (\tfrac{1}{2} - \zeta)(\ell_k(\bar{y}) - \ell_k(x_k)) < 0, \tag{4.11}$$

where the last inequality follows from (4.10). Therefore by continuity, the value of this function for any x in some neighborhood of $\bar{y}$ is negative, implying that (4.8) is satisfied by any approximate solution $\hat{x}$ sufficiently close to $\bar{y}$. □

Theorem 4.6 *Suppose that* $H_k = \nabla^2 f(x_k)$ *in Algorithm 2.1, and that we modify Step 2 in that algorithm to require that the approximate solution* $\hat{x}$ *satisfies* (4.8) *instead of* (2.6). *If we assume that* $\nabla^2 f(x)$ *is Lipschitz continuous, then for all* k *sufficiently large we have* $\alpha_k = 1$.

Proof. Given that $\hat{x} = x_k + d_k$ satisfies (4.8), it follows from Taylor's theorem, the Lipschitz continuity of $\nabla^2 f(x)$, and equation (3.4) that for some constant $\rho > 0$

$$\begin{aligned}
\phi(x_k + d_k) - \phi(x_k) &= [\phi(x_k + d_k) - \phi(x_k) - q_k(x_k + d_k) + q_k(x_k)] \\
&\quad -(q_k(x_k) - q_k(x_k + d_k)) \\
&\leq -\zeta(\ell_k(x_k) - \ell_k(x_k + d_k)) + \rho\|d_k\|^3 \\
&\leq \theta(\ell_k(x_k + d_k) - \ell_k(x_k)) + (\zeta - \theta)(\ell_k(x_k + d_k) - \ell_k(x_k)) + \rho\|d_k\|^3 \\
&\leq \theta(\ell_k(x_k + d_k) - \ell_k(x_k)) - (\zeta - \theta)\frac{\lambda}{2}\|d_k\|^2 + \rho\|d_k\|^3 \\
&\leq \theta(\ell_k(x_k + d_k) - \ell_k(x_k))
\end{aligned}$$

if $\|d_k\| \leq (\zeta - \theta)\lambda/2\rho$. Since the global convergence analysis implies $\|d_k\| \to 0$, we have from (2.7) that eventually the steplength $\alpha_k = 1$ is accepted and used. □

We note that (4.8) is stronger than (2.6), and therefore, all the results presented in this and the previous section apply also to the strengthened condition (4.8). Theorem 4.6 implies that if Algorithm 2.1 is run with the strengthened accuracy condition (4.8), and $H_k = \nabla^2 f(x_k)$, then once the iterates are close enough to a nonsingular minimizer x^*, the iterates have the linear, superlinear or quadratic convergence rates described in Theorem 4.4 if η_k is chosen appropriately.

5 Numerical Results

One of the goals of this paper is to investigate whether the successive quadratic approximation (SQA) method is, in fact, an effective approach for solving convex ℓ_1 regularized problems of the form (1.1). Indeed, it is reasonable to ask whether it might be more effective to apply an algorithm such as ISTA or FISTA, directly to problem (1.1), rather than performing an inner iteration on the subproblem (2.1). Note that each iteration of FISTA requires an evaluation of the gradient of the objective (1.1), whereas each inner iteration for the subproblem (2.1) involves the product of H_k times a vector.

To study this question, we explore various algorithmic options within the successive quadratic approximation method, and evaluate their performance using data sets with different characteristics. One of the data sets concerns the covariance selection problem (where the unknown is a matrix), and the other involves a logistic objective function (where the unknown is a vector). Our benchmark is FISTA applied directly to problem (1.1). FISTA enjoys convergence guarantees when applied to problem (1.1), and is generally regarded as an effective method.

We employ two types of methods for solving the subproblem (2.1) in the successive quadratic approximation method: FISTA and an orthant based method (OBM) [1, 7, 8]. The orthant based method (described in detail in section 5.3) is a two-phase method in which an *active orthant face* of $\mathbb{R}^n$ is first identified, and a subspace minimization is then performed with respect to the variables that define the orthant face. The subspace phase

can be performed by means of a Newton-CG iteration, or by computing a quasi-Newton step; we consider both options.

The methods employed in our numerical tests are as follows.

FISTA. This is the FISTA algorithm [4] applied to the original problem (1.1). We used the implementation from the TFOCS package, called N83 [5]. This implementation differs from the (adaptive) algorithm described by Beck and Teboulle [4] in the way the Lipschitz parameter is updated, and performed significantly better in our test set than the method in [4].

PNOPT. This is the sequential quadratic approximation (proximal Newton) method of Lee, Sun and Saunders [13]. The Hessian H_k in the subproblem (2.1) is updated using the limited memory BFGS formula, with a (default) memory of 50. (The PNOPT package also allows for the use of the exact Hessian, but since this matrix must be formed and factored at each iteration, its use is impractical.) The subproblem (2.1) is solved using the N83 implementation of FISTA mentioned above. PNOPT provides the option of using SPARSA [27] instead of N83 as an inner solver, but the performance of SPARSA was not robust in our tests, and we will not report results with it.

SQA. Is the sequential quadratic approximation method described in Algorithm 2.1. We implemented 3 variants that differ in the method used to solve the subproblem (2.1).

SQA-FISTA. This is an SQA method using FISTA-N83 to solve the subproblem (2.1). The matrix H_k is the exact Hessian $\nabla^2 f(x_k)$; each inner FISTA iteration requires two multiplications with H_k.

SQA-OBM-CG. This is an SQA method that employs an orthant based method to solve the subproblem (2.1). The OBM method performs the subspace minimization step using a Newton-CG iteration. The number of CG iterations varies during the course of the (outer) iteration according to the rule $\min\{3, 1+\lfloor k/10 \rfloor\}$, where k is the outer iteration number.

SQA-OBM-QN. This is an SQA method where the inner solver is an OBM method in which the subspace phase consists of a limited memory BFGS step, with a memory of 50. The correction pairs used to update the quasi-Newton matrix employ gradient differences from the outer iteration (as in PNOPT).

The initial point was set to the zero vector in all experiments, and the iteration was terminated if $\|F(x_k)\|_\infty \leq 10^{-5}$, where F is defined in (2.4). The maximum number of outer iterations for all solvers was 3000. In the SQA method, the parameter η_k in the inexactness condition (2.6) was defined as $\eta_k = \max\{1/k, 0.1\}$, and we set $\theta = 0.1$ in (2.7). For PNOPT we set 'ftol'=$1e-16$, and 'xtol'=$1e-16$ (so that those two tests do not terminate the iteration prematurely), and chose 'Lbfgs_mem'=50.

We noted above that Algorithm 2.1 can employ the inexactness conditions (2.6) or (4.8). We implemented both conditions, with $\zeta = \theta = 0.1$, and obtained identical results in all our runs.

We now describe the numerical tests performed with these methods.

5.1 Inverse Covariance Estimation Problems

The task of estimating a high dimensional sparse inverse covariance matrix is closely tied to the topic of Gaussian Markov random fields [22], and arises in a variety of recognition tasks. This model can be used to recover a sparse social or genetic network from user or experimental data.

A popular approach to solving this problem [2, 3] is to minimize the negative log likelihood function, under the assumption of normality, with an additional ℓ_1 term to enforce sparsity in the estimated inverse covariance matrix. We can write the optimization problem as

$$\min_{P \in \mathbb{R}^{n \times n}} \ \operatorname{tr}(SP) - \log\det P + \mu \|P\|, \tag{5.1}$$

where S is a given sample covariance matrix, P denotes the unknown inverse covariance matrix, μ is the regularization parameter, and $\|P\| \stackrel{\text{def}}{=} \|vec(P)\|_1$. We note that the Hessian of the first two terms in (5.1) has a very special structure: it is given by $P^{-1} \otimes P^{-1}$.

Since the objective is not defined when $\det(P) \leq 0$, we define it as $+\infty$ in that case to ensure that all iterates remain positive definite. Such a strategy could, however, be detrimental to a solver like FISTA, and to avoid this we selected the starting point so that the condition $\det(P) \leq 0$ did not occur.

We employ three data sets: the well-known `Estrogen` and `Leukemia` test sets [14], and the problem given in Olsen et al. [18], which we call `OONR`. The characteristics of the data sets are given in Table 1, where $\operatorname{nnz}(P^*_\mu)$ denotes the number of nonzeros in the solution.

Table 1.

Data set	number of features	μ	$\operatorname{nnz}(P^*_\mu)$
Estrogen	692	0.5	10,614 (2.22%)
Leukemia	1255	0.5	34,781 (2.21%)
OONR	500	0.5	1856 (0.74%)

The performance of the algorithms on these three test problems is given in Tables 2, 3 and 4. We note that FISTA does not perform inner iterations since it is applied directly to the original problem (1.1), and that PNOPT-FISTA does not compute Hessian vector products because the matrix H_k in the model (2.1) is defined by quasi-Newton updating. Each inner iteration of SL-OBM-QN performs a Hessian-vector multiplication to compute the subproblem objective, and a multiplication of the inverse Hessian times a vector to compute the unconstrained minimizer on the active orthant face — we report these as two Hessian-vector products in Tables 2, 3 and 4.

Table 2. ESTROGEN; $\mu = 0.5$, optimality tolerance = 10^{-5}

solver	FISTA	SQA	PNOPT	SQA	SQA
inner solver		FISTA	FISTA	OBM-QN	OBM-CG
outer iterations	808	9	43	44	8
inner iterations	-	183	2134*	64	93
function/gradient evals	1751	10	44	45	10
Hessian-vect mults	-	417	-	2	213
time (s)	208.87	51.54	355.15	38.74	26.95

* For PNOPT we report the number of prox. evaluations

Table 3. LEUKEMIA; $\mu = 0.5$, optimality tolerance = 10^{-5}

solver	FISTA	SQA	PNOPT*	SQA*	SQA
inner solver		FISTA	FISTA	OBM-QN	OBM-CG
outer iterations	838	8	> 488**	101	8
inner iterations	-	187	-	196	101
function/gradient evals	1803	9	-	103	9
Hessian-vect mults	-	420	-	4	239
time (s)	1048.77	239.23	-	171.41	140.33

* out of memory for memory size = 50, we decrease memory size to 5

** exit with message: "Relative change in function value below ftol"

** optimality error is below $1e-4$ after iteration 73, it is $2.3136e-05$ at termination

Table 4. OONR; $\mu = 0.5$, optimality tolerance = 10^{-5}

solver	FISTA	SQA	PNOPT	SQA	SQA
inner solver		FISTA	FISTA	OBM-QN	OBM-CG
outer iterations	212	10	39	37	7
inner iterations	–	80	761	37	60
function/gradient evals	461	11	41	44	9
Hessian-vect mults	–	193	-	2	125
time (s)	23.53	10.14	70.37	12.73	7.09

We now comment on the results given in Tables 2-4. For the inverse covariance selection problem (5.1), Hessian-vector products are not as expensive as for other problems (c.f. Tables 5-6) — in fact, these products are not much costlier than computations with the limited memory BFGS matrix. This fact, combined with the effectiveness of the OBM method, makes SQA-OBM-CG the most efficient of the methods tested. OBM is a good subproblem solver due to its ability to estimate the set of zero variables quickly, so that the subspace step is computed in a small reduced space (the density of P_μ^* is less than 2.5% for the three test problems.) In addition, the OBM-CG method can decrease $\|F_q\|$ drastically in a single iteration, often yielding a high quality SQA step and thus a low number of outer iterations.

We note that the quasi-Newton algorithms SL-OBM-QN and PNOPT are different methods because of the subproblem solvers they employ. SL-OBM-QN uses the two-phase OBM method

in which the quasi-Newton step is computed in a subspace, whereas PNOPT applies the FISTA iteration to subproblem (1.2) where H_k is a quasi-Newton matrix. Although the number of outer iterations of both methods is comparable for problems `Estrogen` and `OONR`, there is a large difference in the number of inner iterations due to power of the OBM approach.

Note that the number of inner FISTA iterations in SQA-FISTA is always smaller than for FISTA. We repeated the experiment with problem `OONR` using looser optimality tolerances (TOL); the total number of FISTA is given in Table 5.

Table 5. Effect of convergence tolerance TOL; OONR

TOL	10^{-2}	10^{-3}	10^{-4}
FISTA (# of outer iterations)	30	74	136
SQA-FISTA (# of inner iterations)	47	56	69

These results are typical for the covariance selection problems, where the SQA-FISTA is clearly more efficient than FISTA; we will see that this is not the case for the problems considered next.

5.2 Logistic Regression Problems

In our second set of experiments the function f in (1.1) is given by a logistic function. Given N data pairs (z_i, y_i), with $z_i \in \mathbb{R}^n$, $y_i \in \mathbb{R}$, $i = 1, \ldots, N$, the optimization problem is given by

$$\min_x \quad \frac{1}{N} \sum_{i=1}^{N} \log(1 + \exp(-y_i x^T z_i)) + \mu \|x\|_1.$$

We employed the data given in Table 6, which was downloaded from the SVMLib repository. The values of the regularization parameter μ were taken from Lee et al. [13].

Table 6. Test problems for logistic regression tests

Data set	N	number of features	μ	nnz(x^*_μ)
Gisette (scaled)	6,000	5,000	1/1500	482 (9.64%)
RCV1 (multi-class)	15,564	47,236	1/62256	144 (0.31%)

Table 7. GISETTE; $\mu = 1/1500$, optimality tolerance = 10^{-5}

solver	FISTA	SQA	PNOPT	SQA	SQA
inner solver		FISTA	FISTA	OBM-QN	OBM-CG
outer iterations	1023	11	237	253	10
inner iterations	–	1744	25260*	1075	770
function/gradient evals	2200	12	240	254	11
Hessian-vector mults	–	3761	–	2	3321
time	185.55	311.28	108.84	38.47	273.10

* For PNOPT we report the number of prox. evaluations.

Table 8. RCV1; $\mu = 3.3065e-04$, optimality tolerance $= 10^{-5}$

solver	FISTA	SQA	PNOPT	SQA	SQA
inner solver		FISTA	FISTA	OBM-QN	OBM-CG
outer iterations	90	7	19	18	6
inner iterations	–	366	1148	27	54
function/gradient evals	184	8	20	19	7
Hessian-vector mults	–	738	–	2	120
time (s)	1.95	7.52	11.23	0.92	1.33

For the logistic regression problems, Hessian-vector products are expensive, particularly for `gisette`, where the data set is dense. As a result, the OBM variant that employs quasi-Newton approximations, namely SQA-OBM-QN, performs best (even though SQA-OBM-CG requires a smaller number of outer iterations). Note that SQA-FISTA is not efficient; in fact it requires a much larger number of inner iterations than the total number of iterations in FISTA. In Table 9 we observe the effect of the optimality tolerance, on these two methods, using problem `gisette`.

Table 9. Effect of convergence tolerance TOL ; Gisette

TOL	10^{-4}	10^{-5}	10^{-6}
FISTA (# of outer iterations)	605	1023	2555
SQA-FISTA (# of inner iterations)	1249	1744	2002

We observe from Table 9, that FISTA requires a smaller number of iterations; it is only for a very high accuracy of 10^{-6} that SQA-FISTA becomes competitive. This is in stark contrast with Table 5.

In summary, for the logistic regression problems the advantage of the SQA method is less pronounced than for the inverse covariance estimation problems, and is achieved only through the appropriate choice of model Hessian H_k (quasi-Newton) and the appropriate choice of inner solver (active set OBM method).

5.3 Description of the orthant based method (OBM)

We conclude this section by describing the orthant-based method used in our experiments to solve the subproblem (2.1). We let t denote the iteration counter of the OBM method, and let z_t denote its iterates.

Given an iterate z_t, the method defines an orthant face Ω_t of $\mathbb{R}^n$ by

$$\Omega_t = \mathbf{cl}(\{d \in \mathbb{R}^n : \mathrm{sgn}(d_i) = \mathrm{sgn}([\omega_t]_i),\ i = 1, ..., n\}), \tag{5.2}$$

with

$$[\omega_t]_i = \begin{cases} \mathrm{sgn}([z_t]_i) & \text{if } [z_t]_i \neq 0 \\ \mathrm{sgn}(-[v_t]_i) & \text{if } [z_t]_i = 0, \end{cases} \tag{5.3}$$

where v_t is the minimum norm subgradient of q_k computed at z_t, i.e.,

$$[v_t]_i = \begin{cases} [\nabla q_k(z_t)]_i + \mu & \text{if } [z_t]_i > 0 \quad \text{or} \quad ([z_t]_i = 0 \ \wedge \ \nabla q_k(z_t)]_i + \mu < 0) \\ [\nabla q_k(z_t)]_i - \mu & \text{if } [z_t]_i < 0 \quad \text{or} \quad ([z_t]_i = 0 \ \wedge \ \nabla q_k(z_t)]_i - \mu > 0) \\ 0 & \text{if } [z_t]_i = 0 \quad \text{and} \quad 0 \in [\nabla q_k(z_t)]_i - \mu, \nabla q_k(z_t)]_i + \mu]. \end{cases} \tag{5.4}$$

Defining Ω_t in this manner was proposed, among others, by Andrew and Gao [1]. In the relative interior of Ω_t, the model function q_k is differentiable. The active set in the orthant-based method, defined as $A^k = \{i : \omega_i^k = 0\}$, determines the variables that are kept at zero, while the rest of the variables are chosen to minimize a (smooth) quadratic model. Specifically, the search direction d_t of the algorithm is given by $d_t = \hat{z} - z_t$, where $\hat{z}$ is a solution of

$$\begin{aligned} \min_{z \in \mathbb{R}^n} \quad & \psi(z) = q_k(z_t) + (z - z_t)^T v^k + \tfrac{1}{2}(z - z_t)^T H_k (z - z_t) \\ \text{s.t.} \quad & z_i = [z_t]_i, \ i \in A^k. \end{aligned} \tag{5.5}$$

Note that $\psi(z) = f(x_k) + (g(x_k) + \omega_t \mu)^T (z - x_k) + \frac{1}{2}(z - x_k)^T H_k (z - x_k)$.

In the OBM-CG variant, we set $H_k = \nabla^2 f(x_k)$, and perform an approximate minimization of this problem using the projected conjugate gradient iteration [17]. In the OBM-QN version, H_k is a limited memory BFGS matrix and $\hat{z}$ is the exact solution of (5.5). This requires computation of the inverse reduced Hessian $R_k = (Z_k^T \nabla^2 H_k Z_k)^{-1}$, where Z_k is a basis for the space defined by (5.5). The matrix R_k can be updated using the compact representations of quasi-Newton matrices [6]. After the direction $d_t = \hat{z} - z_t$ has been computed, the OBM method performs a line search along d_t, projecting the iterate back onto the orthant face Ω_t, until a sufficient reduction in the function q_k has been obtained. Although this algorithm performed reliably in our tests, its convergence has not been proved (to the best of our knowledge) because the orthant face identification procedure (5.2)-(5.5) can lead to arbitrarily small steps.

Our OBM-QN algorithm differs from the OWL method in two respects: it does not realign the direction $z - z_t$ so that the sign of its components match those of v_t, and it performs the minimization of the model exactly, while the OWL method computes only an approximate solution – defined by computing the reduced inverse Hessian $Z_k^T \nabla^2 H_k^{-1} Z_k$, instead of the inverse of the reduced Hessian R_k.

6 Final Remarks

One of the key ingredients in making the successive quadratic approximation (or proximal Newton) method practical for problem (1.1) is the ability to terminate the inner iteration as soon as a step of sufficiently good quality is computed. In this paper, we have proposed such an inexactness criterion; it employs an optimality measure that is tailored to the structure of the problem. We have shown that the resulting algorithm is globally convergent, that its rate of convergence can be controlled through an inexactness parameter, and that the inexact method will naturally accept unit step lengths in a neighborhood of the solution.

We have also argued that our inexactness criterion is preferable to the one proposed by Lee et al. [13].

The method presented in this paper can use any algorithm for the inner minimization of the subproblem (1.2). In particular, all the results are applicable to the case when this inner minimization is performed using a coordinate descent algorithm [12, 25]. In our numerical tests we employed FISTA and an orthant-based method as the inner solvers, and found the latter method to be particularly effective. The efficacy of the successive quadratic approximation approach depends of the choice of matrix H_k in (1.2), which is problem dependent: when Hessian-vector products are expensive to compute, then a quasi-Newton approximation is most efficient; otherwise defining H_k as the exact Hessian and implementing a Newton-CG iteration is likely to give the best results.

Acknowledgement. The authors thank Jong-Shi Pang for his insights and advice throughout the years. The theory presented by Facchinei and Pang [11] in the context of variational inequalities was used in our analysis, showing the power and generality that masterful book.

References

[1] G. Andrew and J. Gao. Scalable training of L_1-regularized log-linear models. In *Proceedings of the 24th international conference on Machine Learning*, pages 33–40. ACM, 2007.

[2] O. Banerjee, L. El Ghaoui, and A. d'Aspremont. Model selection through sparse maximum likelihood estimation for multivariate gaussian or binary data. *The Journal of Machine Learning Research*, 9:485–516, 2008.

[3] O. Banerjee, L. El Ghaoui, A. d'Aspremont, and G. Natsoulis. Convex optimization techniques for fitting sparse Gaussian graphical models. In *Proceedings of the 23rd international conference on Machine learning*, pages 89–96. ACM, 2006.

[4] 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.

[5] Stephen R Becker, Emmanuel J Candès, and Michael C Grant. Templates for convex cone problems with applications to sparse signal recovery. *Mathematical Programming Computation*, 3(3):165–218, 2011.

[6] R. H. Byrd, J. Nocedal, and R. Schnabel. Representations of quasi-newton matrices and their use in limited memory methods. *Mathematical Programming*, 63(4):129–156, 1994.

[7] Richard H Byrd, Gillian M Chin, Jorge Nocedal, and Yuchen Wu. Sample size selection in optimization methods for machine learning. *Mathematical programming*, 134(1):127–155, 2012.

[8] Byrd, R., G. M Chin, J. Nocedal and F. Oztoprak. A family of second-order methods for convex L1 regularized optimization. Technical report, Optimization Center Report 2012/2, Northwestern University, 2012.

[9] R. S. Dembo, S. C. Eisenstat, and T. Steihaug. Inexact-Newton methods. *SIAM Journal on Numerical Analysis*, 19(2):400–408, 1982.

[10] Asen L Dontchev and RT Rockafellar. Convergence of inexact newton methods for generalized equations. *Mathematical Programming*, pages 1–23, 2013.

[11] F. Facchinei and J.S. Pang. *Finite-dimensional variational inequalities and complementarity problems*, volume 1. Springer Verlag, 2003.

[12] C. J. Hsieh, M. A. Sustik, P. Ravikumar, and I. S. Dhillon. Sparse inverse covariance matrix estimation using quadratic approximation. *Advances in Neural Information Processing Systems (NIPS)*, 24, 2011.

[13] Jason D Lee, Yuekai Sun, and Michael A Saunders. Proximal newton-type methods for minimizing composite functions. *arXiv preprint arXiv:1206.1623*, 2012.

[14] L. Li and K. C. Toh. An inexact interior point method for L1-regularized sparse covariance selection. *Mathematical Programming Computation*, 2(3):291–315, 2010.

[15] N. Le Roux M. W. Schmidt and F. Bach. Convergence rates of inexact proximal-gradient methods for convex optimization. *NIPS*, pages 1458–1466, 2011.

[16] A. Milzarek and M. Ulbrich. A semismooth newton method with multi-dimensional filter globalization for l1-optimization. Technical report, Technical University, Munich, 2010.

[17] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer Series in Operations Research. Springer, 1999.

[18] Peder Olsen, Figen Oztoprak, Jorge Nocedal, and Steven Rennie. Newton-like methods for sparse inverse covariance estimation. In *Advances in Neural Information Processing Systems 25*, pages 764–772, 2012.

[19] J. M. Ortega and W. C. Rheinboldt. *Iterative Solution of Nonlinear Equations in Several Variables*. Academic Press, London, 1970.

[20] M. Patriksson. *Nonlinear Programming and Variational Inequality Problems, a Unified Approach*. Kluwer Academic Publishers, Dordrecht, The Netherlands, 1998.

[21] Michael Patriksson. Cost approximation: a unified framework of descent algorithms for nonlinear programs. *SIAM Journal on Optimization*, 8(2):561–582, 1998.

[22] J. D. Picka. Gaussian Markov random fields: theory and applications. *Technometrics*, 48(1):146–147, 2006.

[23] S. Salzo and S. Villa. Inexact and accelerated proximal point algorithms. *Journal of Convex Analysis*, 19(4), 2012.

[24] S. Sra, S. Nowozin, and S.J. Wright. *Optimization for Machine Learning*. Mit Pr, 2011.

[25] X. Tan and K. Scheinberg. Complexity of inexact proximal newton method. Technical report, Dept of ISE, Lehigh University, 2013.

[26] Rachael Tappenden, Peter Richtárik, and Jacek Gondzio. Inexact coordinate descent: complexity and preconditioning. *arXiv preprint arXiv:1304.5530*, 2013.

[27] 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.