An Efficient Method for Large-Scale ℓ_1-Regularized Convex Loss Minimization

Kwangmoo Koh
Department of Electrical Eng.
Stanford University
Stanford, CA 94305-9510, USA
Email: deneb1@stanford.edu

Seung-Jean Kim
Department of Electrical Eng.
Stanford University
Stanford, CA 94305-9510, USA
Email: sjkim@stanford.edu

Stephen Boyd
Department of Electrical Eng.
Stanford University
Stanford, CA 94305-9510, USA
Email: boyd@stanford.edu

Abstract— **Convex loss minimization with ℓ_1 regularization has been proposed as a promising method for feature selection in classification (*e.g.*, l_1-regularized logistic regression) and regression (*e.g.*, l_1-regularized least squares). In this paper we describe an efficient interior-point method for solving large-scale ℓ_1-regularized convex loss minimization problems that uses a preconditioned conjugate gradient method to compute the search step. The method can solve very large problems. For example, the method can solve an l_1-regularized logistic regression problem with a million features and examples (*e.g.*, the 20 Newsgroups data set), in a few minutes, on a PC.**

I. Introduction

A. l_1-regularized convex loss minimization

We consider a problem of the form

$$\text{minimize} \quad (1/m)\sum_{i=1}^{m}\phi(w^Ta_i + vb_i + c_i) + \lambda\|w\|_1, \qquad (1)$$

where the variables are $v \in \mathbf{R}$ and $w \in \mathbf{R}^n$, the problem data are $c_i \in \mathbf{R}$, $a_i \in \mathbf{R}^n$, and $b_i \in \mathbf{R}$, $\lambda > 0$ is the regularization parameter, and $\|w\|_1$ is the ℓ_1 norm of w. Here $\phi : \mathbf{R} \to \mathbf{R}$ is a convex function. In classification and regression setting (which will be described below), ϕ has the meaning of loss and so is called a loss function. The first term in the objective

$$l_{\text{avg}}(v, w) = (1/m)\sum_{i=1}^{m}\phi(w^Ta_i + vb_i + c_i)$$

is called the average loss. This problem is called an ℓ_1-regularized convex loss minimization problem (CLMP).

We refer to the number of nonzero components in w as its *cardinality*, denoted $\mathbf{card}(w)$. Compared with l_2-regularized convex loss minimization (which uses the ℓ_2 penalty function instead of the ℓ_1 penalty, ℓ_1-regularized convex loss minimization tends to yield w with $\mathbf{card}(w)$ small; the regularization parameter λ roughly controls $\mathbf{card}(w)$, with larger λ typically yielding smaller $\mathbf{card}(w)$.

A lot of problems that arise in the context of feature or model selection in signal processing and statistics have the form (1). The main motivation of ℓ_1 regularization is that solving (1) typically yields a *sparse* vector w, *i.e.*, w typically has relatively few nonzero coefficients. (In contrast, ℓ_2 regularization typically yields w with all coefficients nonzero.)

1) l_1-regularized regression: Let $x \in \mathbf{R}^n$ denote a vector of explanatory or feature variables, and $y \in \mathbf{R}$ denote the associated output or outcome. A linear model predicts the output as

$$\hat{y} = w^Tx + v,$$

where $v \in \mathbf{R}$ is the intercept and $w \in \mathbf{R}^n$ is the weight vector.

Suppose we are given a set of (observed or training) examples, $(x_i, y_i) \in \mathbf{R}^n \times \mathbf{R}$, $i = 1, \dots, m$. When the number of features n is greater than the number of examples m, an effective method for finding the weight coefficients and intercept is to solve the l_1-regularized optimization problem

$$\text{minimize} \quad (1/m)\sum_{i=1}^{m}\phi(w^Tu_i + v - y_i) + \sum_{i=1}^{n}\lambda|w_i|, \qquad (2)$$

with variables $v \in \mathbf{R}$ and $w \in \mathbf{R}^n$. Typically ϕ is a symmetric convex function, with $\phi(0) = 0$, such as the quadratic loss, Huber loss, and ϵ-sensitive loss. This problem has the form (1) with $a_i = x_i$, $b_i = 1$, and $c_i = -y_i$.

When the loss function is quadratic, *i.e.*, $\phi(u) = u^2$, the convex loss minimization problem (2) is the l_1-regularized least squares problem that has been studied extensively in the literature (see, *e.g.*, [15], [39]). This feature selection method is called the Lasso [40]. The theoretical properties of the Lasso have been studied by several researchers; see, *e.g.*, [22], [30], [51], [50].

In signal processing, the idea of l_1 regularization arises in the context of sparse signal recovery. The CLMP with the quadratic loss function is related to *compressed sensing* [11], [43] or *compressive sampling* [4], which has many applications in image and signal processing [46], [47], [28], [26], [27]. The effectiveness of l_1 regularization has been studied by many researchers; see, *e.g.*, [5], [6], [7], [12], [13], [14], [16], [17], [19], [41], [42].

2) ℓ_1-regularized classification: Let $x \in \mathbf{R}^n$ denote a vector of explanatory or feature variables, and $y \in \{-1, +1\}$ denote the associated binary output or outcome. Suppose we are given a set of (observed or training) examples, $(x_i, y_i) \in \mathbf{R}^n \times \{-1, +1\}$, $i = 1, \dots, m$.

A classifier which has the form

$$\psi(x) = \mathbf{sgn}(w^Tx + v), \qquad (3)$$

where

$$\operatorname{sgn}(z) = \begin{cases} +1 & z > 0 \\ -1 & z \leq 0, \end{cases}$$

is called linear, since the boundary between the two decision outcomes is a hyperplane (defined by $w^T x + v = 0$).

The weights w and intercept v can be found by solving a problem of the form

$$\text{minimize} \quad (1/m) \sum_{i=1}^{m} \phi(y_i(w^T x_i + v)) + \sum_{i=1}^{n} \lambda |w_i|, \qquad (4)$$

with variables $v \in \mathbf{R}$ and $w \in \mathbf{R}^n$. This problem has the form (1) with $a_i = y_i x_i$, $b_i = y_i$, and $c_i = 0$. The loss function ϕ is small and positive for positive arguments, and grows for negative arguments (*e.g.*, the hinge loss and logistic loss) [20]. The aforementioned method for finding w and v can outperform ℓ_2-regularized classification algorithms, especially when the number of observations is smaller than the number of features.

When ϕ is the logistic loss function ϕ,

$$\phi(z) = \log(1 + \exp(-z)),$$

(4) is a ℓ_1-regularized logistic regression problem. More recently, *ℓ_1-regularized logistic regression* has received much attention; see, *e.g.*, [25], [34].

B. Existing generic solution methods

To solve the ℓ_1-regularized CLMP (1), generic methods for nondifferentiable convex problems can be used, such as the ellipsoid method or subgradient methods [38], [35]. These methods are usually very slow in practice, however. (Because ℓ_1-regularized CLMP typically results in a weight vector with (many) zero components, we cannot simply ignore the nondifferentiability of the objective in the ℓ_1-regularized CLMP (1), hoping to not encounter points of nondifferentiability.)

Faster methods are based on transforming the problem to an equivalent one, with linear inequality constraints,

$$\begin{array}{ll} \text{minimize} & l_{\text{avg}}(v, w) + \lambda \mathbf{1}^T u \\ \text{subject to} & -u_i \leq w_i \leq u_i, \quad i = 1, \ldots, n, \end{array} \qquad (5)$$

where the variables are the original ones $v \in \mathbf{R}$, $w \in \mathbf{R}^n$, along with $u \in \mathbf{R}^n$. Here $\mathbf{1}$ denotes the vector with all components one, so $\mathbf{1}^T u$ is the sum of the components of u. The reformulated problem (5) is a convex optimization problem, with linear constraint functions. When the loss function is twice differentiable, it can be solved by standard convex optimization methods such as SQP, augmented Lagrangian, interior-point, and other methods. High quality solvers that can directly handle the problem (5) include for example LOQO [45], LANCELOT [8], MOSEK [31], and NPSOL [18]. These general purpose solvers can solve small and medium scale ℓ_1-regularized CLMPs quite effectively.

C. Outline

The main goal of this paper is to describe a specialized interior-point method for solving the ℓ_1-regularized convex loss minimization problem (1) where the loss function is twice differentiable. The method uses a preconditioned conjugate gradient approach to compute the search direction and so is a truncated Newton interior-point method.

In Section II, we give (necessary and sufficient) optimality conditions, and a dual problem, for the ℓ_1-regularized CLMP. Using the dual problem, we show how to compute a lower bound on the suboptimality of any pair (v, w). In Section II-C, we describe the truncated Newton interior-point method. In Section IV, we apply the method to l_1-regularized logistic regression. The method can solve very large problems, with a million features and examples (*e.g.*, the 20 Newsgroups data set [24]), in under an hour, on a PC, provided the data matrix is sufficiently sparse.

II. Preliminaries

In this section, we give some preliminaries needed to develop the truncated Newton interior-point method in §II-C.

A. Optimality conditions

The objective function of the ℓ_1-regularized CLMP, $l_{\text{avg}}(v, w) + \lambda \|w\|_1$, is convex but not differentiable, so we use a first-order optimality condition based on subdifferential calculus (see, *e.g.*, [1, Prop. B.24] or [2, §2]). The necessary and sufficient conditions for (v, w) to be optimal for (1) are

$$\begin{array}{rcl} b^T \nu & = & 0, \\ (A^T \nu)_i & \in & \begin{cases} \{-\lambda\} & w_i > 0, \\ \{+\lambda\} & w_i < 0, \quad i = 1, \ldots, n, \\ [-\lambda, \lambda] & w_i = 0, \end{cases} \end{array}$$

where $A = [a_1 \ \cdots \ a_m]^T \in \mathbf{R}^{m \times n}$ and

$$\nu_i = (1/m) \phi'(w^T a_i + v b_i + c_i).$$

These constraints can be expressed as

$$b^T p(v, w) = 0, \qquad (6)$$

and

$$\begin{array}{l} (1/m) \left(A^T p(v, w)\right)_i \\ \in \begin{cases} \{-\lambda\} & w_i > 0, \\ \{+\lambda\} & w_i < 0, \quad i = 1, \ldots, n. \\ [-\lambda, \lambda] & w_i = 0, \end{cases} \end{array} \qquad (7)$$

Here

$$p(v, w) = \begin{bmatrix} \phi'(w^T a_1 + v b_1 + c_1) \\ \vdots \\ \phi'(w^T a_m + v b_m + c_m) \end{bmatrix}.$$

We analyze when a pair of the form $(v, 0)$ is optimal. This occurs if and only if

$$b^T p(v, 0) = 0, \qquad \|(1/m) A^T p_{\log}(v, 0)\|_\infty \leq \lambda.$$

The equation $b^T p(v, 0) = 0$ has the unique solution, say, $\bar{v}$:

$$b^T p(\bar{v}, 0) = 0. \qquad (8)$$

Using this value of v, the second condition becomes

$$\lambda \geq \lambda_{\max} = \|(1/m)A^T p(\bar{v}, 0))\|_\infty. \tag{9}$$

The number $\lambda_{\max}$ gives us an upper bound on the useful range of the regularization parameter λ: for $\lambda \geq \lambda_{\max}$, we get a maximally sparse weight vector, *i.e.*, one with $\mathbf{card}(w) = 0$.

B. *The dual problem*

To derive a Lagrange dual of the ℓ_1-regularized CLMP (1), we first introduce a new variable $z \in \mathbf{R}^m$, as well as new equality constraints $z_i = w^T a_i + vb_i + c_i$, $i = 1, \ldots, m$, to obtain the equivalent problem

$$\begin{array}{ll} \text{minimize} & (1/m)\sum_{i=1}^m \phi(z_i) + \lambda\|w\|_1 \\ \text{subject to} & z_i = w^T a_i + vb_i + c_i, \quad i = 1, \ldots, m. \end{array} \tag{10}$$

Associating dual variables $\nu_i \in \mathbf{R}$ with the equality constraints, the Lagrangian is

$$\begin{aligned} &L(v, w, z, \nu) \\ &\quad = \quad (1/m)\sum_{i=1}^m \phi(z_i) + \lambda\|w\|_1 + \nu^T(-z + Aw + bv + c). \end{aligned}$$

The dual function is

$$\begin{aligned} &\inf_{v,w,z} L(v, w, z, \nu) \\ &\quad = \begin{cases} -(1/m)\sum_{i=1}^m \phi^*(-m\nu_i) + \nu^T c & \|A^T\nu\|_\infty \leq \lambda, \\ & b^T\nu = 0, \\ -\infty & \text{otherwise}, \end{cases} \end{aligned}$$

where ϕ^* is the *conjugate* of the convex loss function ϕ:

$$\phi^*(y) = \sup_{u \in \mathbf{R}} \left(yu - \phi(u)\right). \tag{11}$$

For general background on convex duality and conjugates, see, *e.g.*, [3, Chap. 5] or [2].

The dual of the problem (1) is

$$\begin{array}{ll} \text{maximize} & -(1/m)\sum_{i=1}^m \phi^*(-m\nu_i) + \nu^T c \\ \text{subject to} & \|A^T\nu\|_\infty \leq \lambda, \quad b^T\nu = 0, \end{array} \tag{12}$$

where

$$G(\nu) = -(1/m)\sum_{i=1}^m \phi^*(-m\nu_i) + \nu^T c, \tag{13}$$

$A = [a_1 \ \cdots \ a_m]^T \in \mathbf{R}^{m \times n}$, the variable is $\nu \in \mathbf{R}^m$, and ϕ^* is the conjugate of the loss function ϕ,

$$\phi^*(y) = \sup_{u \in \mathbf{R}} \left(yu - \phi(u)\right).$$

The dual problem (12) is a convex optimization problem with variable $\nu \in \mathbf{R}^m$, and has the form of an ℓ_∞-norm constrained maximum generalized entropy problem. We say that $\nu \in \mathbf{R}^m$ is *dual feasible* if it satisfies $\|A^T\nu\|_\infty \leq \lambda$, $b^T\nu = 0$.

From standard results in convex optimization we have the following.

- *Weak duality.* Any dual feasible point ν gives a lower bound on the optimal value p^* of the (primal) ℓ_1-regularized CLMP (1):
$$G(\nu) \leq p^\star. \tag{14}$$
- *Strong duality.* The ℓ_1-regularized CLMP (1) satisfies a variation on Slater's constraint qualification, so there is an optimal solution $\nu^\star$ of the dual (12), which satisfies
$$G(\nu^\star) = p^\star.$$
In other words, the optimal values of the primal (1) and dual (12) are equal.

C. *Suboptimality bound*

We can derive a bound on the suboptimality of (v, w), by constructing a dual feasible point $\bar{\nu}$, from an arbitrary w,

$$\bar{\nu} = (s/m)p(\bar{v}, w) \tag{15}$$

where $\bar{v}$ is the optimal intercept for the offset w,

$$\bar{v} = \arg\min_v (1/m)\sum_{i=1}^m \phi(w^T a_i + vb_i + c_i), \tag{16}$$

and the scaling constant s is given by $s = \min\left\{m\lambda/\|A^T p(\bar{v}, w))\|_\infty, 1\right\}$.

The difference between the primal objective value of (v, w), and the associated lower bound $G(\bar{\nu})$, is called the *duality gap*, and denoted $\eta(v, w)$:

$$\eta(v, w) = l_{\text{avg}}(v, w) + \lambda\|w\|_1 - G(\bar{\nu}). \tag{17}$$

We always have $\eta(v, w) \geq 0$; and (by weak duality (14)) the point (v, w) is no more than η-suboptimal. At the optimal point $(v^\star, w^\star)$, we have $\eta = 0$.

D. *The logarithmic barrier and central path*

The *logarithmic barrier* for the bound constraints $-u_i \leq w_i \leq u_i$ in (5) is

$$\Phi(w, u) = -\sum_{i=1}^n \log(u_i + w_i) - \sum_{i=1}^n \log(u_i - w_i),$$

with domain

$$\mathbf{dom}\,\Phi = \{(w, u) \in \mathbf{R}^n \times \mathbf{R}^n \mid |w_i| < u_i, \ i = 1, \ldots, n\}.$$

The logarithmic barrier function is smooth and convex. We augment the weighted objective function by the logarithmic barrier, to obtain

$$\phi_t(v, w, u) = tl_{\text{avg}}(v, w) + t\lambda\mathbf{1}^T u + \Phi(w, u), \tag{18}$$

where $t > 0$ is a parameter. This function is smooth, strictly convex, and bounded below, and so has a unique minimizer which we denote $(v^\star(t), w^\star(t), u^\star(t))$. This defines a curve in $\mathbf{R} \times \mathbf{R}^n \times \mathbf{R}^n$, parametrized by t, called the *central path*. (See [3, §11] for more on the central path and its properties.)

With the point $(v^\star(t), w^\star(t), u^\star(t))$ we associate

$$\nu^\star(t) = p(v^\star(t), w^\star(t)), \tag{19}$$

which can be shown to be dual feasible. (Indeed, it coincides with the dual feasible point $\bar{\nu}$ constructed from $w^\star(t)$ using the method of Section II-C.) The associated duality gap satisfies

$$l_{\text{avg}}(v^\star(t), w^\star(t)) + \lambda\|w^\star(t)\|_1 - G(\nu^\star(t)) \leq 2n/t.$$

In other words, $(v^\star(t), w^\star(t))$ is no more than $2n/t$-suboptimal, so the central path leads to an optimal solution. (See [3, Chap. 11] for more on the central path.)

III. An interior-point method

In this section we describe an interior-point method for solving the ℓ_1-regularized CLMP (1), in the equivalent formulation given in (5).

In a primal interior-point method, we compute a sequence of points on the central path, for an increasing sequence of values of t, using Newton's method to minimize $\phi_t(v, w, u)$, starting from the previously computed central point. A typical method uses the sequence $t = t_0, \mu t_0, \mu^2 t_0, \ldots$, where μ is between 2 and 50 (see, *e.g.*, [3, §11.3]). The method can be terminated when $2n/t \leq \epsilon$, since then we can guarantee ϵ-suboptimality of $(v^\star(t), w^\star(t))$. See, *e.g.*, [32], [48], [49] for more on (primal) interior-point methods.

Using our method for cheaply computing a dual feasible point and associated duality gap for *any* (v, w) (and not just for (v, w) on the central path, as in the general case), we can construct a custom interior-point method that updates the parameter t at each iteration. This method is an extension of the interior-point method for ℓ_1-regularized logistic regression developed in [23].

```
Custom Interior-point Method.
given tolerance $\epsilon > 0$, $\alpha \in (0, 1/2)$, $\beta \in (0, 1)$
Set initial values.
$t := 1/\lambda$, $v := \bar{v}$, the solution of (8), $w := 0$, $u := 1$.
repeat
    1. Compute search direction.
       Solve the Newton system
       $\nabla^2 \phi_t(v, w, u) \begin{bmatrix} \Delta v \\ \Delta w \\ \Delta u \end{bmatrix} = -\nabla \phi_t(v, w, u)$.
    2. Backtracking line search.
       Find the smallest integer $k \geq 0$ that satisfies
       $\phi_t(v + \beta^k \Delta v, w + \beta^k \Delta w, u + \beta^k \Delta u)$
           $\leq \phi_t(v, w, u) + \alpha \beta^k \nabla \phi_t(v, w, u)^T \begin{bmatrix} \Delta v \\ \Delta w \\ \Delta u \end{bmatrix}$.
    3. Update. $(v, w, u) := (v, w, u) + \beta^k (\Delta v, \Delta w, \Delta u)$.
    4. Set $v := \bar{v}$, the optimal value of the intercept, as in (16).
    5. Construct dual feasible point $\nu$ from (15).
    6. Evaluate duality gap $\eta$ from (17).
    7. quit if $\eta \leq \epsilon$.
    8. Update $t$.
```

The computational effort per iteration is dominated by step 1, the search direction computation. When the search direction in Newton's method is computed approximately, using an iterative method such as a preconditioned conjugate gradient (PCG) method, the overall algorithm is called a *conjugate gradient Newton method*, or a *truncated Newton method* [37], [9]. (Truncated Newton methods have been applied to interior-point methods; see, *e.g.*, [44], [36].)

A. Update rule and parameters

We describe the choice of initial values for v, w, u, and t. The choice $v = \bar{v}$ is the optimal value of v when $w = 0$ and $u = \mathbf{1}$. The choice $w = 0$, $u = \mathbf{1}$, and $t = 1/\lambda$ seems to work very well, especially when the original data are standardized (*i.e.*, a_i have zero mean and unit variance). (In any case, the choice of the initial values does not greatly affect performance.) The construction of a dual feasible point and duality gap, in steps 4–6, is explained in Section II-C. Typical values for the line search parameters are $\alpha = 0.01$, $\beta = 0.5$, but here too, these parameter values do not have a large effect on performance.

There are many possible update rules for the parameter t. In a classical primal barrier method, t is held constant until ϕ_t is (approximately) minimized, *i.e.*, $\|\nabla \phi_t\|_2$ is small; when this occurs, t is increased by a factor typically between 2 and 50. More sophisticated update rules can be found in, *e.g.*, [32], [48], [49].

The update rule we propose is

$$t := \begin{cases} \max\left\{\mu \min\{\hat{t}, t\}, t\right\}, & s \geq s_{\min} \\ t, & s < s_{\min} \end{cases} \qquad (20)$$

where $\hat{t} = 2n/\eta$, and $s = \beta^k$ is the step length chosen in the line search. Here $\mu > 1$ and $s_{\min} \in (0, 1]$ are algorithm parameters; we have found good performance with $\mu = 2$ and $s_{\min} = 0.5$. This update rule uses the step length s as a crude measure of proximity to the central path. We can also give an informal justification of convergence of the custom interior-point algorithm [23]. (A formal proof of convergence would be quite long.)

B. Hessian and gradient

In this section we give explicit formulas for the gradient and Hessian of ϕ_t. The gradient $g = \nabla \phi_t(v, w, u)$ is given by

$$g = \begin{bmatrix} g_1 \\ g_2 \\ g_3 \end{bmatrix} \in \mathbf{R}^{2n+1},$$

where

$$\begin{aligned} g_1 &= \nabla_v \phi_t(v, w, u) \\ &= (t/m) b^T p(v, w), \\ g_2 &= \nabla_w \phi_t(v, w, u) \\ &= (t/m) A^T p(v, w) + \begin{bmatrix} 2w_1/(u_1^2 - w_1^2) \\ \vdots \\ 2w_n/(u_n^2 - w_n^2) \end{bmatrix}, \\ g_3 &= \nabla_u \phi_t(v, w, u) \\ &= t\lambda \mathbf{1} - \begin{bmatrix} 2u_1/(u_1^2 - w_1^2) \\ \vdots \\ 2u_n/(u_n^2 - w_n^2) \end{bmatrix}. \end{aligned}$$

The Hessian $H = \nabla^2 \phi_t(v, w, u)$ is given by

$$H = \begin{bmatrix} tb^T D_0 b & tb^T D_0 A & 0 \\ tA^T D_0 b & tA^T D_0 A + D_1 & D_2 \\ 0 & D_2 & D_1 \end{bmatrix} \in \mathbf{R}^{(2n+1)\times(2n+1)},$$

where

$$\begin{aligned} D_0 &= \frac{1}{m}\operatorname{diag}(\phi''(w^Ta_1+vb_1+c_1),\ldots,\phi''(w^Ta_m+vb_m+c_m)),\\ D_1 &= \operatorname{diag}\left(\frac{2(u_1^2+w_1^2)}{(u_1^2-w_1^2)^2},\ldots,\frac{2(u_n^2+w_n^2)}{(u_n^2-w_n^2)^2}\right),\\ D_2 &= \operatorname{diag}\left(\frac{-4u_1w_1}{(u_1^2-w_1^2)^2},\ldots,\frac{-4u_nw_n}{(u_n^2-w_n^2)^2}\right). \end{aligned}$$

Here, we use $\operatorname{diag}(z_1,\ldots,z_m)$ to denote the diagonal matrix with diagonal entries $z_1,\ldots,z_m$, where $z_i \in \mathbf{R}$, $i=1,\ldots,m$. The Hessian H is symmetric and positive definite.

The search direction is defined by the linear equations (Newton system)

$$\begin{bmatrix} tb^TD_0b & tb^TD_0A & 0\\ tA^TD_0b & tA^TD_0A+D_1 & D_2\\ 0 & D_2 & D_1 \end{bmatrix}\begin{bmatrix}\Delta v\\ \Delta w\\ \Delta u\end{bmatrix} = -\begin{bmatrix} g_1\\ g_2\\ g_3\end{bmatrix}.$$

We first eliminate Δu to obtain the reduced Newton system

$$H_{\text{red}}\begin{bmatrix}\Delta v\\ \Delta w\end{bmatrix} = -g_{\text{red}}, \qquad (21)$$

where

$$\begin{aligned} H_{\text{red}} &= \begin{bmatrix} tb^TD_0b & tb^TD_0A\\ tA^TD_0b & tA^TD_0A+D_3\end{bmatrix},\\ g_{\text{red}} &= \begin{bmatrix} g_1\\ g_2-D_2D_1^{-1}g_3\end{bmatrix},\\ D_3 &= D_1-D_2D_1^{-1}D_2. \end{aligned}$$

Once this reduced system is solved, Δu can be recovered as

$$\Delta u = -D_1^{-1}(g_3+D_2\Delta w).$$

Several methods can be used to solve the reduced Newton system (21), depending on the relative sizes of n and m and the sparsity of the data A.

C. Computing the search direction

The PCG algorithm [10, §6.6] computes an approximate solution of the Newton syste, $Hx=-g$. It uses a preconditioner $P \in \mathbf{R}^{2n\times 2n}$, also symmetric positive definite.

Preconditioned Conjugate Gradients Algorithm

```
given relative tolerance $\epsilon_{\text{pcg}} > 0$, iteration limit $N_{\text{pcg}}$,
    and $x_0 \in \mathbf{R}^{2n}$
$k := 0$, $r_0 := Hx_0 - g$, $p_1 := -P^{-1}g$, $y_0 := P^{-1}r_0$.
repeat
    $k := k+1$
    $z := Hp_k$
    $\nu_k := y_{k-1}^Tr_{k-1}/p_k^Tz$
    $x_k := x_{k-1} + \nu_kp_k$
    $r_k := r_{k-1} - \nu_kz$
    $y_k := P^{-1}r_k$
    $\mu_{k+1} := y_k^Tr_k/y_{k-1}^Tr_{k-1}$
    $p_{k+1} := y_k + \mu_{k+1}p_k$
until $\|r_k\|_2/\|g\|_2 \le \epsilon_{\text{pcg}}$ or $k = N_{\text{pcg}}$.
```

Each iteration of the PCG algorithm involves a handful of inner products, the matrix-vector product Hp_k and a solve step with P in computing $P^{-1}r_k$. With exact arithmetic, and ignoring the stopping condition, the PCG algorithm is guaranteed to compute the exact solution $x=-H^{-1}g$ in N steps. When $P^{-1/2}HP^{-1/2}$ is well conditioned, or has just a few extreme eigenvalues, the PCG algorithm can compute an approximate solution in a number of steps that can be far smaller than N. Since $P^{-1}r_k$ is computed in each step, we need this computation to be efficient.

The truncated Newton interior-point method is the same as the interior-point algorithm described in Section III, with the search direction computed using the PCG algorithm.

We can compute Hp_k in the PCG algorithm as

$$\begin{aligned} Hp_k &= \begin{bmatrix} tb^TD_0b & tb^TD_0A & 0\\ tA^TD_0b & tA^TD_0A+D_1 & D_2\\ 0 & D_2 & D_1\end{bmatrix}\begin{bmatrix}p_{k1}\\ p_{k2}\\ p_{k3}\end{bmatrix}\\ &= \begin{bmatrix} b^Tu\\ A^Tu+D_1p_{k2}\\ D_2p_{k2}+D_1p_{k3}\end{bmatrix}, \end{aligned}$$

where $u = tD_0(bp_{k1}+Ap_{k2}) \in \mathbf{R}^m$. The cost of computing Hp_k is $O(p)$ flops when A is sparse with p nonzero elements. (We assume $p \ge n$, which holds if each example has at least one nonzero feature.)

We now describe a simple choice for the preconditioner P. The Hessian can be written as

$$H = t\nabla^2 l_{\text{avg}}(v,w) + \nabla^2\Phi(w,u).$$

To obtain the preconditioner, we replace the first term with its diagonal part, to get

$$\begin{aligned} P &= \operatorname{diag}\left(t\nabla^2 l_{\text{avg}}(v,w)\right) + \nabla^2\Phi(w,u)\\ &= \begin{bmatrix} d_0 & 0 & 0\\ 0 & D_3 & D_2\\ 0 & D_2 & D_1\end{bmatrix}, \end{aligned} \qquad (22)$$

where

$$d_0 = tb^TD_0b, \qquad D_3 = \operatorname{diag}(tA^TD_0A) + D_1.$$

(Here $\operatorname{diag}(S)$ is the diagonal matrix obtained by setting the off-diagonal entries of the matrix S to zero.) This preconditioner approximates the Hessian of tl_{avg} with its diagonal entries, while retaining the Hessian of the logarithmic barrier. For this preconditioner, $P^{-1}r_k$ can be computed cheaply as

$$\begin{aligned} P^{-1}r_k &= \begin{bmatrix} d_0 & 0 & 0\\ 0 & D_3 & D_2\\ 0 & D_2 & D_1\end{bmatrix}^{-1}\begin{bmatrix} r_{k1}\\ r_{k2}\\ r_{k3}\end{bmatrix}\\ &= \begin{bmatrix} r_{k1}/d_0\\ (D_1D_3-D_2^2)^{-1}(D_1r_{k2}-D_2r_{k3})\\ (D_1D_3-D_2^2)^{-1}(-D_2r_{k2}+D_3r_{k3})\end{bmatrix}, \end{aligned}$$

which requires $O(n)$ flops.

There are several good choices for the initial point in the PCG algorithm (labeled x_0), such as the negative gradient, or the previous search direction. We have found good performance with both, with a small advantage in using the previous search direction.

The PCG relative tolerance parameter ϵ_{pcg} has to be carefully chosen to obtain good efficiency in a truncated Newton method. If the tolerance is too small, too many PCG steps are need to compute each search direction; if the tolerance is too high, then the computed search directions do not give adequate reduction in duality gap per iteration. The adaptive rule

$$\epsilon_{\text{pcg}} = \min\{0.1, \xi\eta/\|g\|_2\}, \tag{23}$$

where g is the gradient and η is the duality gap at the current iterate, appears to give good results for a wide range of problems with $\xi = 0.3$. In other words, we solve the Newton system with low accuracy (but never worse than 10%) at early iterations, and solve it more accurately as the duality gap decreases. This adaptive rule is similar in spirit to standard methods used in inexact and truncated Newton methods; see, *e.g.*, [33].

The computational effort of the truncated Newton interior-point algorithm is the product of s, the total number of PCG steps required over all iterations, and the cost of a PCG step, which is $O(p)$, where p is the number of nonzero entries in A, *i.e.*, the total number of (nonzero) features appearing in all examples. In extensive testing, we found the truncated Newton interior-point method to be very efficient, requiring a total number of PCG steps ranging between a few hundred (for medium size problems) and several thousand (for large problems). For medium size (and sparse) problems it was faster than the interior-point method that uses direct methods to solve the Newton system; moreover the truncated Newton interior-point method was able to solve very large problems, for which forming the Hessian H (let alone computing the search direction) would be prohibitively expensive.

While the total number of iterations in the interior-point method that uses direct methods to solve the Newton system is around 35, and nearly independent of the problem size and problem data, the total number of PCG iterations required by the truncated Newton interior-point method can vary significantly with problem data and the value of the regularization parameter λ. In particular, for small values of λ (which lead to large values of $\mathbf{card}(w)$), the truncated Newton interior-point method requires a larger total number of PCG steps. Algorithm performance that depends substantially on problem data, as well as problem dimension, is typical of all iterative (*i.e.*, non direct) methods, and is the price paid for the ability to solve very large problems.

IV. l_1-REGULARIZED LOGISTIC REGRESSION EXAMPLES

In this section we demonstrate the performance of the interior-point method described in Section III with some l_1-regularized logistic regression examples. We use algorithm parameters

$$\alpha = 0.01, \quad \beta = 0.5, \quad s_{\min} = 0.5, \quad \epsilon = 10^{-8}.$$

$\lambda/\lambda_{\max}$	$\mathbf{card}(w)$	Iterations	PCG iterations	Time (sec)
0.5	9	43	558	134
0.1	544	60	1036	256
0.05	2531	58	2090	501

I: Performance of truncated Newton interior-point method on the 20 newsgroup data set ($n = 777811$ features, $m = 11314$ examples) for 3 values of λ.

(The algorithm performs well for much smaller values of ϵ, but this accuracy is more than adequate for any practical use.) We chose the parameter N_{pcg} to be large enough (5000) that the iteration limit was never reached in our experiments; the typical number of PCG iterations was far smaller. The algorithm is implemented in both Matlab and C, on a 3.2GHz Pentium IV running Linux. The C implementation is available online (`www.stanford.edu/~boyd/reports/l1_logreg.html`).

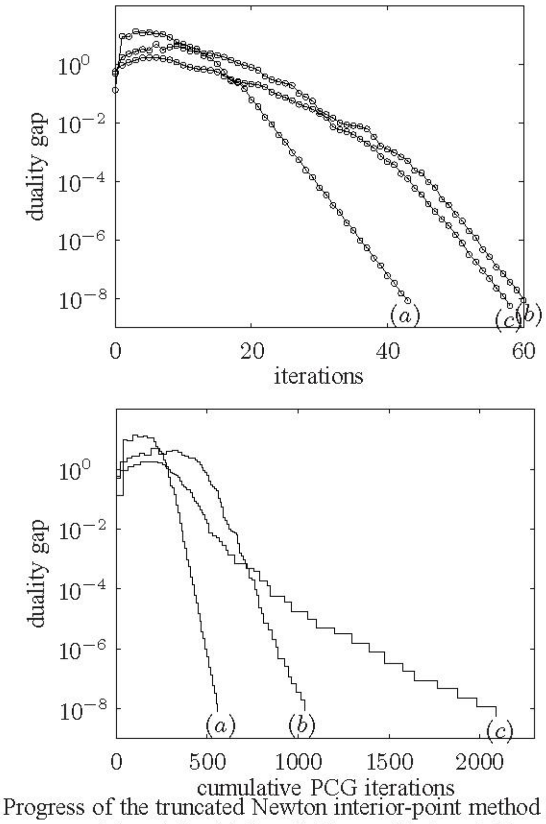
A. A large problem

We use the 20 Newsgroups data set [24]. We processed the data set in a way similar to [21]. The positive class consists of the 10 groups with names of form sci.*, comp.*, and misc.forsale, and the negative class consists of the other 10 groups. We used McCallum's Rainbow program [29] with the command

```
rainbow -g 3 -h -s -O 2 -i
```

to tokenize the (text) data set. These options specify trigrams, skip message headers, no stoplist, and drop terms occurring fewer than two times. The resulting data set has $n = 777811$ features (trigrams) and $m = 11314$ examples (articles). Each example contains an average of 425 nonzero features. The total number of nonzero entries in the data matrix A is $p = 4802169$. We standardized the data set using implicit standardization, as explained in Section III-C, solving three ℓ_1-regularized LRPs, with $\lambda = 0.5\lambda_{\max}$, $\lambda = 0.1\lambda_{\max}$, and $\lambda = 0.05\lambda_{\max}$. (For the value $\lambda = 0.01\lambda_{\max}$, the runtime is on the order of one hour. This case is not of practical interest, and so not reported here, since the cardinality of the optimal solution is around 10000 and comparable to the number of examples.) The performance of the algorithm, and the cardinality of the weight vectors, is given in table I. Figure 1 shows the progress of the algorithm, with duality gap versus iteration (lefthand plot), and duality gap versus cumulative PCG iteration (righthand plot).

The number of iterations required to solve the problems ranges between 43 and 60, depending on λ. The more relevant measure of computational effort is the total number of PCG iterations, which ranges between around 500 and 2000, again, increasing with decreasing λ, which corresponds to increasing $\mathbf{card}(w)$. The average number of PCG iterations, per iteration of the truncated Newton interior-point method, is around 13 for $\lambda = 0.5\lambda_{\max}$, 17 for $\lambda = 0.1\lambda_{\max}$, and 36 for $\lambda = 0.05\lambda_{\max}$. (The variance in the number of PCG iterations required per iteration, however, is large.) The running time is consistent with a cost of around 0.24 seconds per PCG iteration. The



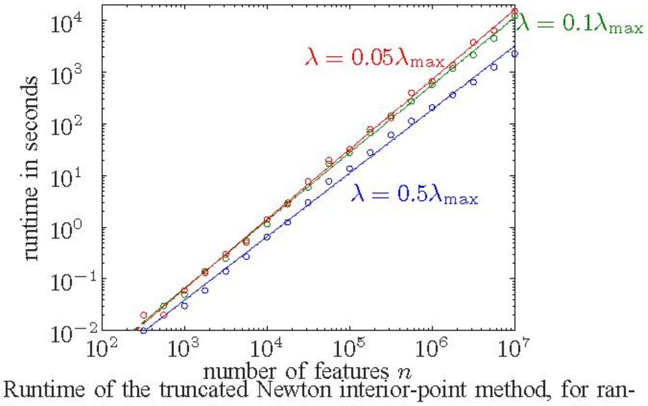
1: Progress of the truncated Newton interior-point method on the 20 Newsgroups data set for (a) $\lambda = 0.5\lambda_{\max}$, (b) $\lambda = 0.1\lambda_{\max}$, and (c) $\lambda = 0.05\lambda_{\max}$. *Top.* Duality gap versus iterations. *Bottom.* Duality gap versus cumulative PCG iterations.

increase in running time, for decreasing λ, is due primarily to an increase in the average number of PCG iterations required per iteration, but also significantly from an increase in the overall number of iterations required.

B. Randomly generated problems

To examine the effect of problem size on the runtime of the truncated Newton interior-point method, we generated a family of 21 data sets, with the number of features n varying from one hundred to ten millions, and $m = 0.1n$ examples. Each problem has an equal number of positive and negative examples. Features of positive (negative) examples are independent and identically distributed, drawn from a normal distribution $\mathcal{N}(v, 1)$, where v is in turn drawn from a uniform distribution on $[0, 1]$ for positive examples ($[-1, 0]$ for negative examples). In doing so, the sparsity was controlled to have the average number of nonzero features per example around 30. We standardized the data sets, solving each problem instance for the three values $\lambda = 0.5\lambda_{\max}$, $\lambda = 0.1\lambda_{\max}$, and $\lambda = 0.05\lambda_{\max}$.

The total runtime, for the 63 ℓ_1-regularized logistic regression problems, is shown in figure 2. The plot shows that runtime increases as λ decreases, and grows approximately linearly with problem size.



2: Runtime of the truncated Newton interior-point method, for randomly generated sparse ℓ_1-regularized logistic regression problems, with three values of λ.

Acknowledgments

This work is supported in part by the National Science Foundation under grants #0423905 and (through October 2005) #0140700, by the Air Force Office of Scientific Research under grant #F49620-01-1-0365, by MARCO Focus center for Circuit & System Solutions contract #2003-CT-888, and by MIT DARPA contract #N00014-05-1-0700. The authors thank Michael Grant, Trevor Hastie, Honglak Lee, Suin Lee, Mee Young Park, Yinyu Ye, and Sungroh Yoon for helpful comments and suggestions.

References

[1] D. Bertsekas. *Nonlinear Programming*. Athena Scientific, second edition, 1999.

[2] J. Borwein and A. Lewis. *Convex Analysis and Nonlinear Optimization*. Springer, 2000.

[3] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[4] E. Candès. Compressive sampling. In *Proceedings of International Congress of Mathematics*, 2006.

[5] E. Candès and J. Romberg. Quantitative robust uncertainty principles and optimally sparse decompositions. *Foundations of Computational Mathematics*, 6(2):227–254, 2006.

[6] E. Candès, J. Romberg, and T. Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Information Theory*, 52(2):489–509, 2006.

[7] E. Candès and T. Tao. Near optimal signal recovery from random projections: universal encoding strategies? *IEEE Transactions on Information Theory*, 52(12):5406–5425, 2006.

[8] A. Conn, N. Gould, and Ph. Toint. *LANCELOT: A Fortran package for large-scale nonlinear optimization (Release A)*, volume 17 of *Springer Series in Computational Mathematics*. Springer-Verlag, 1992.

[9] R. Dembo and T. Steihaug. Truncated-Newton algorithms for large-scale unconstrained optimization. *Math. Program.*, 26:190–212, 1983.

[10] J. Demmel. *Applied Numerical Linear Algebra*. Society for Industrial and Applied Mathematics, 1997.

[11] D. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306, 2006.

[12] D. Donoho. For most large underdetermined systems of linear equations, the minimal l_1-norm solution is also the sparsest solution. *Communications on Pure and Applied Mathematic*, 59(6):797–829, 2006.

[13] D. Donoho and M. Elad. Optimally sparse representation in general (non-orthogonal) dictionaries via ℓ^1 minimization. *Proc. Nat. Aca. Sci.*, 100(5):2197–2202, March 2003.

[14] D. Donoho and X. Huo. Uncertainty principle and ideal atomic decomposition. *IEEE Transactions on Information Theory*, 47(7):2845–2862, 2001.

[15] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *Annals of Statistics*, 32(2):407–499, 2004.

[16] M. Elad and A. Bruckstein. A generalized uncertainty principle and sparse representation in pairs of bases. *IEEE Transactions on Information Theory*, 48(9):2558–2567, 2002.

[17] A. Feuer and A. Nemirovski. On sparse representation in pairs of bases. *IEEE Transactions on Information Theory*, 49(6):1579–1581, 2003.

[18] P. Gill, W. Murray, M. Saunders, and M. Wright. User's guide for NPSOL (Version 4.0): A FORTRAN package for nonlinear programming. Technical Report SOL 86-2, Operations Research Dept., Stanford University, Stanford, California 94305, January 1986.

[19] R. Gribonval and M. Nielsen. Sparse representation in unions of bases. *IEEE Transactions on Information Theory*, 48(12):3320–3325, 2003.

[20] T. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning*. Springer Series in Statistics. Springer-Verlag, New York, 2001.

[21] S. Keerthi and D. DeCoste. A modified finite Newton method for fast solution of large scale linear SVMs. *Journal of Machine Learning Research*, 6:341–361, 2005.

[22] K. Knight and W. Fu. Asymptotics for Lasso-type estimators. *Annals of Statistics*, 28(5):1356–1378, 2000.

[23] K. Koh, S.-J. Kim, and S. Boyd. An interior-point method for ℓ_1-regularized logistic regression, 2006. Manuscript. Available from www.stanford.edu/~boyd/l1_logistic_reg.html.

[24] K. Lang. Newsweeder: Learning to filter netnews. In *Proceedings of the twenty-first international conference on Machine learning (ICML)*, pages 331–339, 1995.

[25] S. Lee, H. Lee, P. Abeel, and A. Ng. Efficient l_1-regularized logistic regression. In *Proceedings of the 21st National Conference on Artificial Intelligence (AAAI-06)*, 2006.

[26] M. Lustig, D. Donoho, and J. Pauly. Rapid MR imaging with compressed sensing and randomly under-sampled 3DFT trajectories. In *Proceedings of the 14th Annual Meeting of ISMRM*, 2006.

[27] M. Lustig, D. Donoho, and J. Pauly. Sparse MRI: The application of compressed sensing for rapid MR imaging, 2007. Manuscript. Available from www.stanford.edu/~mlustig/.

[28] M. Lustig, J. Lee, D. Donoho, and J. Pauly. $k-t$ SPARSE: High frame rate dynamic mri exploiting spatio-temporal sparsity. In *Proceedings of the 14th Annual Meeting of ISMRM*, 2006.

[29] A. McCallum. Bow: A toolkit for statistical language modeling, text retrieval, classification and clustering. Available from www.cs.cmu.edu/~mccallum/bow, 1996.

[30] N. Meinshausen and P. Büuhlmann. High-dimensional graphs and variable selection with the Lasso. *Annals of Statistics*, 34(3):14361462, 2006.

[31] MOSEK ApS. *The MOSEK Optimization Tools Version 2.5. User's Manual and Reference*, 2002. Available from www.mosek.com.

[32] Y. Nesterov and A. Nemirovsky. *Interior-Point Polynomial Methods in Convex Programming*, volume 13 of *Studies in Applied Mathematics*. SIAM, Philadelphia, PA, 1994.

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

[34] M. Park and T. Hastie. An ℓ_1 regularization-path algorithm for generalized linear models, 2006. Manuscript. Available from www-stat.stanford.edu/~hastie/Papers/glmpath.pdf.

[35] B. Polyak. *Introduction to Optimization*. Optimization Software, 1987. Translated from Russian.

[36] L. Portugal, M. Resende, G. Veiga, and J. Júdice. A truncated primal-infeasible dual-feasible network interior point method, 1994.

[37] A. Ruszczynski. *Nonlinear Optimization*. Princeton university press, 2006.

[38] N. Z. Shor. *Minimization Methods for Non-differentiable Functions*. Springer Series in Computational Mathematics. Springer, 1985.

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

[40] R. Tibshirani. The Lasso for variable selection in the Cox model. *Statistics in Medicine*, 16:385–395, 1997.

[41] J. Tropp. Recovery of short, complex linear combinations via l_1 minimization. *IEEE Transactions on Information Theory*, 51(4):1568–1570, 2005.

[42] J. Tropp. Just relax: Convex programming methods for identifying sparse signals in noise. *IEEE Transactions on Information Theory*, 53(3):1030–1051, 2006.

[43] Y. Tsaig and D. Donoho. Extensions of compressed sensing. *Signal Processing*, 86(3):549–571, 2006.

[44] L. Vandenberghe and S. Boyd. A primal-dual potential reduction method for problems involving matrix inequalities. *Math. Program.*, 69:205–236, 1995.

[45] R. Vanderbei. *LOQO User's Manual — Version 3.10*, 1997. Available from www.orfe.princeton.edu/loqo.

[46] M. Wakin, J. Laska, M. Duarte, D. Baron, S. Sarvotham, D. Takhar, K. Kelly, and R. Baraniuk. An architecture for compressive imaging. In *Proceedings of International Conference on Image Processing (ICIP)*, 2006.

[47] M. Wakin, J. Laska, M. Duarte, D. Baron, S. Sarvotham, D. Takhar, K. Kelly, and R. Baraniuk. Compressive imaging for video representation and coding. In *Proceedings of Picture Coding Symposium (PCS)*, 2006.

[48] S. Wright. *Primal-dual interior-point methods*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1997.

[49] Y. Ye. *Interior Point Algorithms: Theory and Analysis*. John Wiley & Sons, 1997.

[50] P. Zhao and B. Yu. On model selection consistency of Lasso. *Journal of Machine Learning Research*, 7:2541–2563, 2006.

[51] H. Zou. The adaptive Lasso and its oracle properties. *Journal of the American Statistical Association*, 101(476):1418–1429, 2006.