In Pursuit of a Root

Ewout van den Berg **Michael P. Friedlander**
Department of Computer Science
University of British Columbia
{ewout78,mpf}@cs.ubc.ca

Abstract

The basis pursuit technique is used to find a minimum one-norm solution of an underdetermined least-squares problem. Basis pursuit denoise fits the least-squares problem only approximately, and a single parameter determines a curve that traces the trade-off between the least-squares fit and the one-norm of the solution. We show that the function that describes this curve is convex and continuously differentiable over all points of interest. The dual solution of a least-squares problem with an explicit one-norm constraint gives function and derivative information needed for a root-finding method. As a result, we can compute arbitrary points on this curve. Numerical experiments demonstrate that our method, which relies on only matrix-vector operations, scales well to large problems.

1 Basis pursuit denoise

Basis pursuit aims to find a sparse solution of the underdetermined system of equations $Ax = b$, where $A \in \mathbb{R}^{m\times n}$ and $b \in \mathbb{R}^m$. Typically, $m \ll n$, and the problem is ill-posed. The approach advocated by Chen et al. [5] is to solve basis pursuit (BP) problem

$$\underset{x\in\mathbb{R}^n}{\text{minimize}} \quad \|x\|_1 \quad \text{subject to} \quad Ax = b. \tag{1}$$

In the presence of noisy or imperfect data, however, it is undesirable to fit the linear system exactly. Instead, the constraint in (1) is relaxed to obtain the basis pursuit denoise (BPDN) problem

$$(\text{BP}_\sigma) \qquad \underset{x}{\text{minimize}} \quad \|x\|_1 \quad \text{subject to} \quad \|Ax - b\|_2 \le \sigma,$$

where $\sigma \ge 0$ is an estimate of the noise level in the data. The case $\sigma = 0$ corresponds to a (BP) solution. There is now a significant body of work that addresses the conditions under which a solution of this problem coincides with the *sparsest* solution of the underdetermined system (see [8, 18] and references therein).

We present an algorithm, suitable for large-scale applications, that is capable of finding an accurate solution of (BP_σ) for any value of $\sigma \ge 0$. Our approach is based on recasting (BP_σ) as a problem of finding the root of a single-variable nonlinear equation. At each iteration of our algorithm, an estimate of that variable is used to define a convex optimization problem whose solution yields derivative information that can be used by a Newton-based root finding algorithm.

The one-norm regularized least-squares problem

$$\underset{x}{\text{minimize}} \quad \tfrac{1}{2}\|Ax - b\|_2^2 + \gamma\|x\|_1 \tag{2}$$

is often called a BPDN problem; this is the problem statement first proposed by [5]. It is well known that, for appropriate choices of σ and γ, the solutions of (BP_σ) and (2) coincide. Formulation (2) is often preferred because of its close connection to convex quadratic programming (see, e.g., [12]), for which many standard algorithms and software are available. However, except for special cases (such

as A orthogonal) the parameters that make these problems equivalent cannot be known a priori, and it is often not clear how to choose the parameter γ.

We focus on the situation where the parameter σ is known (perhaps approximately)—e.g., if there is an estimate of noise level inherent in the underlying system or in the measurements taken. In this case, it is preferable to solve (BP$_\sigma$) directly. In this paper we outline a method for doing this.

1.1 Approach

At the heart of our approach is the ability to efficiently solve the one-norm constrained least-squares problem

$$(\text{LASSO}_\lambda) \qquad \underset{x}{\text{minimize}} \quad \|Ax-b\|_2 \quad \text{subject to} \quad \|x\|_1 \leq \lambda$$

using a spectral projected-gradient (SPG) algorithm [3]. Like (2), this problem is parameterized by a scalar; the crucial difference, however, is that the dual solution of (LASSO$_\lambda$) yields vital information on how to update λ so that the next solution of (LASSO$_\lambda$) is much closer to the solution of (BP$_\sigma$).

Each iteration of the SPG method requires an orthogonal projection of an n-vector onto the convex set $\{x \mid \|x\|_1 \leq \lambda\}$. In §3 we present an $\mathcal{O}(n+k\log n)$ algorithm for this projection, where k is the number of nonzeros in the vector. In many important applications, A is a Fourier-type operator, and matrix-vector products with A and A^T can be obtained with $\mathcal{O}(n\log n)$ cost. The dominant cost in our algorithm, then, consists of the matrix-vector products, as it does in other algorithms for BPDN and (LASSO$_\lambda$).

Let x_λ denote the optimal solution of (LASSO$_\lambda$). The single-parameter function

$$\phi(\lambda) = \|r_\lambda\|_2 \quad \text{with} \quad r_\lambda := b - Ax_\lambda \tag{3}$$

gives the optimal value of (LASSO$_\lambda$) for each $\lambda \geq 0$. As we describe in §2, the derivative of $\phi(\lambda)$ is given by $-\mu_\lambda$, where $\mu_\lambda \geq 0$ is the dual solution of (LASSO$_\lambda$). Importantly, this dual solution can be easily obtained as a by-product of the minimization of (LASSO$_\lambda$). Our approach is thus based on applying Newton's method to find a root of the nonlinear equation

$$\phi(\lambda) = \sigma, \tag{4}$$

which defines a sequence of regularization parameters $\lambda^k \to \lambda^*$, where x_{λ^*} is a solution of (BP$_\sigma$). In §2.3 we present rate-of-convergence results for the case where ϕ and ϕ' are known only approximately, that is, when (LASSO$_\lambda$) is approximately minimized. This is in contrast to the usual inexact-Newton analysis which assumes that ϕ is known exactly.

We make the blanket assumptions that the vector $b \in \text{range}(A)$ and $b \neq 0$. Note that this is only needed to simplify the discussion, and implies that (BP$_\sigma$) is feasible for all $\sigma \geq 0$.

1.2 Related work

A number of approaches have been suggested for solving (BP$_\sigma$), many of which are based on repeatedly solving (2) for various values of γ. Notable examples of this approach are HOMOTOPY [15] and LARS [11], which solve (2) for essentially all values of γ. In this way, they eventually discover the value of γ that recovers a solution of (BP$_\sigma$). These active-set continuation approaches begin with $\gamma = \|A^Tb\|_\infty$ (for which the corresponding solution $x_\gamma = 0$), and gradually reduce γ in stages that predictably change the sparsity pattern in x_γ. The remarkable efficiency of these continuation methods stems from their ability to systematically update the resulting sequence of solutions ([9]). The computational bottleneck is the solution at each iteration of a least-squares subproblem that involves a subset of the columns of A. Moreover, even if the target value γ_σ is known a priori, the methods necessarily begin with $\gamma = \|A^Tb\|_\infty$ and traverse all critical values of γ down to γ_σ.

The problem (BP$_\sigma$) can be considered to be a special case of the generic second-order cone program [4, Ch. 5]. Interior-point (IP) algorithms for general conic programs can be very effective if the matrices are available explicitly. Examples of general-purpose software for conic programs include SeDuMi [17] and MOSEK [13]. The software package SparseLab [10] implements an IP algorithm specially adapted to (BP$_\sigma$). The efficiency of software implementations based on IP algorithms ultimately relies on the ability to efficiently solve certain systems of highly ill-conditioned matrices.

Our application of the SPG algorithm closely follows Birgin et al. [3] in minimizing general nonlinear functions over arbitrary convex sets. Their proposed method combines projected-gradient search directions with the spectral steplength that was introduced by Barzilai and Borwein [1]. The key ingredient of Birgin et al.'s algorithm is the projection of the gradient direction onto a convex set, which in our case is defined by the constraint in (LASSO_λ).

In a recent report, Figueiredo et al. [12] describe the efficiency of an SPG method specialized to (2). Their approach builds on the earlier report by Dai and Fletcher [6] on the efficiency of a specialized SPG method for general, bound-constrained quadratic programs.

2 The optimal trade-off curve

The function ϕ defined by (3) yields the optimal value of the constrained problem (LASSO_λ) for each value of the regularization parameter λ. Its graph traces the optimal trade-off between λ, which constrains $\|x\|_1$, and the residual norm $\|Ax - b\|_2$. Fig. 1(a) shows the graph of ϕ for a typical problem.

To deal with the nondifferentiability of the one-norm constraints, we appeal to Lagrange duality theory. This approach yields significant insight into the properties of the optimal trade-off curve. We discuss the most important properties below.

2.1 The LASSO dual

In order to derive the dual of (LASSO_λ), we first recast it as the equivalent problem

$$\underset{r,x}{\text{minimize}} \quad \|r\|_2 \quad \text{subject to} \quad Ax + r = b, \quad \|x\|_1 \leq \lambda. \tag{5}$$

The dual of this convex problem is given by

$$\underset{y,\mu}{\text{maximize}} \quad \mathcal{L}(y, \mu) \quad \text{subject to} \quad \mu \geq 0, \tag{6}$$

where

$$\mathcal{L}(y, \mu) = \inf_{x,r} \ \|r\|_2 - y^T(Ax + r - b) + \mu(\|x\|_1 - \lambda)$$

is the Lagrange dual function. We use the separability of the infimum in r and x to rearrange terms and arrive at the equivalent statement

$$\mathcal{L}(y, \mu) = b^T y - \lambda\mu - \sup_r \ (y^T r - \|r\|_2) - \sup_x \ (y^T A x - \mu\|x\|_1).$$

We recognize the suprema above as the conjugate functions of $\|r\|_2$ and $\mu\|x\|_1$, respectively. For an arbitrary norm $\|\cdot\|$ with dual norm $\|\cdot\|_*$, the conjugate function of $f(x) = \alpha\|x\|$ for any $\alpha \geq 0$ is given by ([4, §3.3.1])

$$f_*(y) := \sup_x \ (y^T x - \alpha\|x\|) = \begin{cases} 0 & \text{if } \|y\|_* \leq \alpha, \\ \infty & \text{otherwise.} \end{cases} \tag{7}$$

With this expression of the conjugate function, it follows that (6) remains bounded if and only if the dual variables y and μ satisfy the constraints $\|y\|_2 \leq 1$ and $\|A^T y\|_\infty \leq \mu$. The dual of (5) is then given by

$$\underset{y,\mu}{\text{maximize}} \quad b^T y - \lambda\mu \quad \text{subject to} \quad \|y\|_2 \leq 1, \quad \|A^T y\|_\infty \leq \mu. \tag{8}$$

The dual variables y and μ can easily be computed from the optimal primal solutions. To derive y, first note that from (7),

$$\sup_r \ (y^T r - \|r\|_2) = 0 \quad \text{if} \quad \|y\|_2 \leq 1. \tag{9}$$

Therefore $y = r/\|r\|_2$, and we can without loss of generality take $\|y\|_2 = 1$ in (8). To derive μ, note that as long as $\lambda > 0$, μ must be at its lower bound, as implied by the constraint $\|A^T y\|_\infty \leq \mu$, and so we take $\mu = \|A^T y\|_\infty$. (If $r = 0$ or $\lambda = 0$, the choice, respectively, of y and μ is arbitrary.)

The dual variable can then be eliminated, and we arrive at the following necessary and sufficient optimality conditions for the primal-dual solution $(r_\lambda, x_\lambda, \mu_\lambda)$ of (5):

$$\|x_\lambda\|_1 \leq \lambda, \quad \|A^T r_\lambda\|_\infty \leq \mu_\lambda \|r_\lambda\|_2, \quad \text{and} \quad \mu_\lambda(\|x_\lambda\|_1 - \lambda) = 0, \tag{10}$$

which respectively describe primal feasibility, dual feasibility, and complementarity.

2.2 Convexity and differentiability of the trade-off curve

Our root-finding procedure for locating specific points on the optimal trade-off curve relies on several important properties of the function ϕ. Let λ_{BP} be the optimal objective value of the BP problem (1). As we show below, ϕ is nonincreasing, and its definition thus implies that λ_{BP} is the first point at which the graph of ϕ touches the horizontal axis. Our assumption that $0 \neq b \in \text{range}(A)$ implies that (1) is feasible, and $\lambda_{\text{BP}} > 0$. Therefore, at the endpoints of the interval of interest,

$$\phi(0) = \|b\|_2 \quad \text{and} \quad \phi(\lambda_{\text{BP}}) = 0. \tag{11}$$

As the following result confirms, the function is convex, and strictly decreasing over the interval $\lambda \in [0, \lambda_{\text{BP}}]$. Crucially, it is also continuously differentiable on the interval of interest.

Theorem 2.1. (a) *The function ϕ is convex and nonincreasing.* (b) *For all $\lambda \in (0, \lambda_{\text{BP}})$, ϕ is continuously differentiable, $\phi'(\lambda) = -\mu_\lambda$, and the optimal dual variable $\mu_\lambda = \|A^T y_\lambda\|_\infty$, where $y_\lambda = r/\|r_\lambda\|_2$.* (c) *For $\lambda \in [0, \lambda_{\text{BP}}]$, $\|x_\lambda\|_1 = \lambda$, and ϕ is strictly decreasing.*

Proof. (a) The function ϕ can be restated as $\phi(\lambda) = \inf_x f(x, \lambda)$, where

$$f(x, \lambda) := \|Ax - b\|_2 + \psi_\lambda(x) \quad \text{and} \quad \psi_\lambda(x) := \begin{cases} 0 & \text{if } \|x\|_1 \leq \lambda, \\ \infty & \text{otherwise.} \end{cases}$$

Note that by (7), $\psi_\lambda(x) = \sup_z x^T z - \lambda \|z\|_\infty$, which is the pointwise supremum of an affine function in (x, λ). Therefore it is convex in (x, λ). Together with the convexity of $\|Ax - b\|_2$, this implies that f is convex in (x, λ). Consider any $\lambda_1, \lambda_2 \geq 0$, and let x_1 and x_2 be the corresponding minimizers of ϕ. For any $\gamma \in [0, 1]$,

$$\begin{aligned} \phi(\gamma\lambda_1 + (1-\gamma)\lambda_2) &= \inf_x \; f(x, \gamma\lambda_1 + (1-\gamma)\lambda_2) \\ &\leq f(\gamma x_1 + (1-\gamma)x_2, \gamma\lambda_1 + (1-\gamma)\lambda_2) \\ &\leq \gamma f(x_1, \lambda_1) + (1-\gamma) f(x_2, \lambda_2) \\ &= \gamma\phi(\lambda_1) + (1-\gamma)\phi(\lambda_2), \end{aligned}$$

and so ϕ is convex in λ. Also, ϕ is nonincreasing because the feasible set enlarges as λ increases.

(b) The function ϕ is differentiable at λ if and only if its subgradient at λ is unique [16, Theorem 25.1]. By [2, Proposition 6.5.8(a)], $-\mu_\lambda \in \partial\phi(\lambda)$. Therefore, to prove differentiability of ϕ it is enough show that μ_λ is unique. Note that μ appears linearly in (8) with coefficient $-\lambda < 0$, and thus μ_λ is not optimal unless it is at its lower bound, as implied by the constraint $\|A^T y_\lambda\|_\infty$. Hence, $\mu_\lambda = \|A^T y_\lambda\|_\infty$. Moreover, convexity of (LASSO$_\lambda$) implies that its optimal value is unique, and so $r_\lambda \equiv b - Ax_\lambda$ is unique. Also, $\|r_\lambda\| > 0$ because $\lambda < \lambda_{\text{BP}}$. As discussed in connection with (9), we can then take $y_\lambda = r_\lambda/\|r_\lambda\|_2$, and so uniqueness of r_λ implies uniqueness of y_λ and thus uniqueness of μ_λ, as required. The continuity of the gradient follows from the convexity of ϕ.

(c) The assertion holds trivially for $\lambda = 0$, and for $\lambda = \lambda_{\text{BP}}$, $\|x_{\lambda_{\text{BP}}}\| = \lambda_{\text{BP}}$ by definition. It only remains to prove (c) on the interior of the interval. Note that $\phi(\lambda) \equiv \|r_\lambda\| > 0$ for all $\lambda \in [0, \lambda_{\text{BP}})$. Then by (b), $-\mu < 0$ (and hence ϕ is strictly decreasing for $\lambda < \lambda_{\text{BP}}$). But because x_λ and μ_λ both satisfy the complementarity condition in (10), it must hold that $\|x_\lambda\|_1 = \lambda$. □

Interestingly, the technique used to prove convexity and differentiability in Theorem 2.1 does not in any way rely on the specific norm used for the regularization function. Thus, an optimal trade-off curve defined for any p-norms in (LASSO$_\lambda$) has the properties of convexity and differentiability.

2.3 Root finding

As we briefly outlined in §1.1, our algorithm generates a sequence of regularization parameters $\lambda^k \to \lambda^*$ based on the iteration

$$\lambda^{k+1} = \lambda^k + \Delta\lambda^k \quad \text{with} \quad \Delta\lambda^k := -(\phi(\lambda^k) - \sigma)/\phi'(\lambda^k), \tag{12}$$

such that the corresponding solutions $x^k := x_{\lambda^k}$ of (LASSO$_{\lambda^k}$) converge to x^*. For values of $\sigma \in (0, \|b\|_2)$, Theorem 2.1 implies that ϕ is convex, strictly decreasing, and continuously differentiable. In that case it is clear that $\lambda^k \to \lambda^*$ superlinearly for all initial values $\lambda^0 \in (0, \lambda_{\text{BP}})$ (see, e.g., [14]).

The efficiency of our method, as with many Newton-type methods for large problems, ultimately relies on the ability to carry out the iteration described by (12) with only an approximation to $\phi(\lambda^k)$ and $\phi'(\lambda^k)$. Although the nonlinear equation (4) that we wish to solve involves only a single variable λ, the evaluation of $\phi(\lambda)$ involves the solution of (LASSO$_\lambda$), which can be a large optimization problem that is expensive to solve to full accuracy.

For systems of nonlinear equations in general, inexact Newton methods assume that the Newton system analogous to the equation

$$\phi'(\lambda^k)\Delta\lambda^k = -(\phi(\lambda^k) - \sigma)$$

is solved only approximately with a residual that is a fraction of the right-hand side. A constant fraction yields a linear convergence rate, and a fraction tending to zero yields a superlinear convergence rate (see, e.g., [14].) However, the inexact Newton analysis does not apply to the case where the right-hand side (i.e., the function itself) is known only approximately and it is therefore not possible to know a priori the accuracy required to achieve an inexact-Newton-type convergence rate. This is the situation that we are faced with if (LASSO$_\lambda$) is solved approximately. As we show below, with only approximate knowledge of the function value ϕ, this inexact version of Newton's method still converges, though the convergence rate is sublinear.

2.3.1 Approximate primal-dual solutions

The algorithm for solving (LASSO$_\lambda$) that we outline in §3 maintains feasibility of the iterates at all iterations. Thus, an approximate solution $\bar{x}_\lambda$ and its corresponding residual $\bar{r}_\lambda := b - A\bar{x}_\lambda$ satisfy

$$\|\bar{x}_\lambda\|_1 \leq \lambda, \quad \text{and} \quad \|\bar{r}_\lambda\|_2 \geq \|r_\lambda\|_2 > 0, \tag{13}$$

where the second set of inequalities holds because $\bar{x}_\lambda$ is suboptimal and $\lambda < \lambda_{\text{BP}}$. We can thus construct approximations to the dual variables

$$\bar{y}_\lambda = \bar{r}_\lambda / \|\bar{r}_\lambda\|_2 \quad \text{and} \quad \bar{\mu}_\lambda = \|A^T\bar{y}_\lambda\|_\infty \tag{14}$$

which are dual feasible, i.e., they satisfy the second inequality in (10). The value of the dual problem (6) at any feasible point gives a lower bound on the optimal value $\|r_\lambda\|$, and the value of the primal problem (5) at any feasible point gives an upper bound on the optimal value. Therefore,

$$b^T\bar{y}_\lambda - \lambda\bar{\mu}_\lambda \leq \|r_\lambda\| \leq \|\bar{r}_\lambda\|.$$

We use the duality gap

$$\delta_\lambda := \|\bar{r}_\lambda\|_2 - (b^T\bar{y}_\lambda - \lambda\bar{\mu}_\lambda) \geq 0 \tag{15}$$

to measure the quality of the approximate solution $\bar{x}_\lambda$.

Let $\bar{\phi}(\lambda) := \|\bar{r}_\lambda\|_2$ be the objective value of (LASSO$_\lambda$) at the approximate solution $\bar{x}_\lambda$. The duality gap at $\bar{x}_\lambda$ provides a bound on the difference between $\phi(\lambda)$ and $\bar{\phi}(\lambda)$. If we additionally assume that A has full rank (so that the smallest singular value is positive), we can also use δ_λ to provide a bound on the difference between μ_λ and $\bar{\mu}_\lambda$. From Theorem 2.1(b) and from (14)–(15),

$$\bar{\phi}(\lambda) - \phi(\lambda) < \delta_\lambda \quad \text{and} \quad |\bar{\phi}'(\lambda) - \phi'(\lambda)| < \gamma\delta_\lambda \tag{16}$$

for some positive constant γ that is inversely proportional to the smallest singular value of A and is independent of λ.

2.3.2 Local convergence rate

The following theorem establishes the local convergence rate of an inexact Newton method for (4) where ϕ and ϕ' are known only approximately.

Theorem 2.2. *Suppose A has full rank, $\sigma \in (0, \|b\|_2)$, and that $\delta^k := \delta_{\lambda^k} \to 0$. Then if λ^0 is close enough to λ^*, the iteration* (12)*, with ϕ and ϕ' replaced by $\bar{\phi}$ and $\bar{\phi}'$, generates a sequence $\lambda^k \to \lambda^*$ that satisfies*

$$|\lambda^{k+1} - \lambda^*| = \gamma_1\delta^k + \eta^k|\lambda^k - \lambda^*|,$$

where $\eta^k \to 0$ and γ_1 is a positive constant.

Proof. Because $\phi(\lambda^*) = \sigma \in (0, \|b\|_2)$, equation (11) implies that $\lambda^* \in (0, \lambda_{\text{BP}})$. By Theorem 2.1 we have that $\phi(\lambda)$ is continuously differentiable for all λ close enough to λ^*, so, by Taylor's theorem,

$$\begin{aligned}\phi(\lambda^k) - \sigma &= \int_0^1 \phi'(\lambda^* + \alpha[\lambda^k - \lambda^*])\, d\alpha\, (\lambda^k - \lambda^*) \\ &= \phi'(\lambda^k)(\lambda^k - \lambda^*) + \int_0^1 \left[\phi'(\lambda^* + \alpha[\lambda^k - \lambda^*]) - \phi'(\lambda^k)\right]\, d\alpha\, (\lambda^k - \lambda^*) \\ &= \phi'(\lambda^k)(\lambda^k - \lambda^*) + \omega(\lambda^k, \lambda^*),\end{aligned}$$

where the remainder ω satisfies

$$\omega(\lambda^k, \lambda^*)/|\lambda^k - \lambda^*| \to 0 \quad \text{as} \quad |\lambda^k - \lambda^*| \to 0. \tag{17}$$

By (16), and because (13) holds for $\lambda = \lambda^k$, there exist positive constants γ_1 and γ_2 independent of λ^k such that

$$\left| \frac{\phi(\lambda^k) - \sigma}{\phi'(\lambda^k)} - \frac{\bar{\phi}(\lambda^k) - \sigma}{\bar{\phi}'(\lambda^k)} \right| \leq \gamma_1 \delta^k \quad \text{and} \quad |\phi'(\lambda^k)|^{-1} < \gamma_2.$$

Then,

$$\begin{aligned}|\lambda^{k+1} - \lambda^*| &= |\lambda^k - \lambda^* + \Delta\lambda^k| \\ &= \left| -\frac{\bar{\phi}(\lambda^k) - \sigma}{\bar{\phi}'(\lambda^k)} + \frac{1}{\phi'(\lambda^k)} \left[\phi(\lambda^k) - \sigma - \omega(\lambda^k, \lambda^*)\right] \right| \\ &\leq \left| \frac{\phi(\lambda^k) - \sigma}{\phi'(\lambda^k)} - \frac{\bar{\phi}(\lambda^k) - \sigma}{\bar{\phi}'(\lambda^k)} \right| + \left| \frac{\omega(\lambda^k, \lambda^*)}{\phi'(\lambda^k)} \right| \\ &= \gamma_1 \delta^k + \gamma_2 |\omega(\lambda^k, \lambda^*)| \\ &= \gamma_1 \delta^k + \eta^k |\lambda^k - \lambda^*|,\end{aligned}$$

where $\eta^k := \gamma_2 |\omega(\lambda^k, \lambda^*)|/|\lambda^k - \lambda^*|$. With λ^k sufficiently close to λ^*, (17) implies that $\eta^k < 1$. Recursively apply the above inequality $\ell \geq 1$ times to obtain

$$|\lambda^{k+\ell} - \lambda^*| \leq \gamma_1 \sum_{i=1}^{\ell} (\gamma_1)^{\ell - i} \delta^{k+i-1} + (\eta^k)^\ell |\lambda^k - \lambda^*|,$$

and because $\delta^k \to 0$ and $\eta^k < 1$, it follows that $\lambda^{k+\ell} \to y^*$ as $\ell \to \infty$. Hence $\lambda^k \to \lambda^*$, as required. By again applying (17), we have that $\eta^k \to 0$. □

Note that if (LASSO_λ) is solved exactly at each iteration so that $\delta^k = 0$, then Theorem 2.2 shows that the convergence rate is superlinear, as we expect. In effect, the convergence rate of the algorithm depends on the rate at which $\delta^k \to 0$.

3 Spectral gradient projection for the LASSO subproblem

The SPG procedure that we use for solving the subproblem (LASSO_λ) is based on [3, Algorithm 2.1]. Each iteration of the algorithm computes the objective gradient of (LASSO_λ)—requiring a matrix-vector product with A and A^T—and performs a linesearch along the projected gradient direction. Normally, a subsequent projection is needed to evaluate the optimality of the current iterate, but we save one projection step by instead using the duality gap to test optimality.

A core component of the SPG method is the ability to efficiently solve the problem

$$\underset{p}{\text{minimize}} \quad \|c - p\|_2 \quad \text{subject to} \quad \|p\|_1 \leq \lambda, \tag{18}$$

which gives the orthogonal projection of an n-vector c onto a one-norm ball of radius λ. In this section we give an $\mathcal{O}(n + k \log n)$ algorithm, where k is the number of nonzeros in c.

We assume, without loss of generality, that the entries of c are nonnegative. If they are not, we then recast the objective as $\|D(c - p)\|_2$, where $D = \text{diag}\big(\text{sign}(c)\big)$. The true solution is recovered by again applying D.

Our algorithm for solving (18) is motivated as follows. We begin with the trial solution $p \leftarrow c$. If this is feasible for (18), we exit immediately with $p^* = c$. Otherwise, we decrease the norm of p by $\nu := \|c\|_1 - \lambda$, at the expense of increasing the objective. Therefore, we must find a vector d such that $\|p - d\|_1 = \lambda$, and in order to minimize the effect on the objective, choose d so that $\|c - (p - d)\|_2 = \|d\|_2$ is minimal. Therefore, d must solve the problem

$$\underset{d}{\text{minimize}} \quad \|d\|_2 \quad \text{subject to} \quad d \geq 0,\ \|d\|_1 = \nu, \tag{19}$$

which has necessary and sufficient optimality conditions $d = \gamma e$, $e^T d = \nu$, and $\gamma \geq 0$, for some γ. Premultiplying the first condition by e^T and using the second condition, we see that $\gamma = \nu/n$. The solution of (19) is then $d^* = (\nu/n)e$.

Algorithm 1: Projection onto the set $\{x \mid \|x\|_1 \leq \lambda\}$

Input: c, λ **Output**: p
if $\|c\|_1 \leq \lambda$ **then return** c
$\alpha \leftarrow 0,\ \tau \leftarrow 0,\ \nu \leftarrow -\lambda$
$p \leftarrow$ `BuildHeap` (c)
for $j \leftarrow 0$ **to** n **do**
  $p_{\max} \leftarrow p[1]$
  $\nu \leftarrow \nu + p_{\max}$
  $p \leftarrow$ `DeleteMax` (p)
  $\alpha \leftarrow \nu/j$
  if $\alpha \geq p_{\max}$ **then** break
  $\tau \leftarrow \alpha$
$p \leftarrow$ `SoftThreshold` (c, τ)
return p

However, we cannot exit with $p \leftarrow c - d^*$ if some of these entries are negative (the projection must preserve the sign pattern of c). Therefore, if each $d_i^* < c_{\min} := \min_i c_i$, we set $p^* \leftarrow c - d^*$ and exit with the solution of (18). Otherwise, we enforce

$$p_i^* = 0 \quad \text{for all} \quad i \in \mathcal{I} := \{i \mid d_i^* \geq c_{\min}\},$$

and recursively repeat the procedure described above for the remaining variables $\{1, \ldots, n\} \backslash \mathcal{I}$.

Algorithm 1 formally describes this approach, and is based upon a binomial heap structure in which the first element is the largest element in absolute value. The cost of `BuildHeap` is $\mathcal{O}(n)$, whereas that of `DeleteMax`, which extracts the maximal element from the heap and restores the heap property, is $\mathcal{O}(\log n)$. The final step is to apply the soft-threshold operation `SoftThreshold`:

$$p_i \leftarrow c_i - \text{sign}(c_i)\tau \quad \text{if} \quad |c_i| \geq \tau, \qquad \text{or} \qquad p_i \leftarrow 0 \quad \text{otherwise}, \quad i = 1, \ldots, n.$$

The overall projection algorithm is therefore $\mathcal{O}(n + k \log n)$. Note that we can interpret orthogonal projection onto the one-norm ball as the problem of finding a suitable soft-threshold value τ.

4 Numerical experiments

We implement our algorithm in a Matlab package named SPGL1. For the numerical experiments shown in Figs. 1(b)–(d), we generate a matrix A and vector b as described in [12] and consider separate experiments for (LASSO$_\lambda$) and for (BP$_\sigma$). For (LASSO$_\lambda$), we compare SPGL1 against SparseLab [10], soft-thresholding [7], Homotopy [15], and GPSR [12]; see Fig.1(b). SPGL1 outperforms the other methods for large γ in (2), and is only marginally slower than GPSR for lower values. For (BP$_\sigma$), we compare SPGL1 against SeDuMi [17] and SparseLab. To analyze performance and scalability on (BP$_\sigma$), we fix σ and $n = 4m$, and vary m. SeDuMi, being general purpose, is slowest. SparseLab outperforms SPGL1 on smaller problems but does not scale as effectively. To compare these two solvers, we need knowledge of the optimal dual solution μ_λ, as computed by SPGL1.

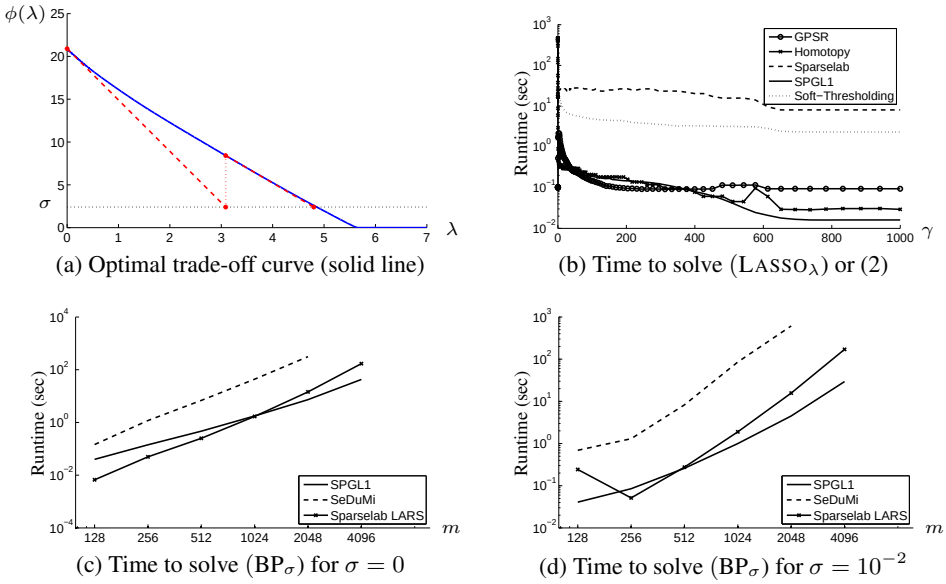
5 Looking ahead

The efficiency of our root-finding algorithm ultimately depends on the efficiency with which the subproblems (LASSO$_\lambda$) can be solved. Although the SPG algorithm outlined in §3 performs well in our tests, there may be room for improvement by applying a Newton-type method. Such an approach opens the door to preconditioning strategies that are not available in the SPG algorithm. Our main motivation for proposing this algorithm is its usefulness for inpainting applications in seismic imaging in geophysics, where the problem sizes easily stretch into millions of variables.

References

[1] J. Barzilai and J. M. Borwein. Two-point step size gradient methods. *IMA J. Numer. Anal.*, 8:141–148, 1988.

[2] D. P. Bertsekas. *Convex Analysis and Optimization*. Athena Scientific, Belmont, MA, 2003.



(a) Optimal trade-off curve (solid line)

(b) Time to solve (LASSO$_\lambda$) or (2)

(c) Time to solve (BP$_\sigma$) for $\sigma = 0$

(d) Time to solve (BP$_\sigma$) for $\sigma = 10^{-2}$

Figure 1: (a) The optimal trade-off curve for a typical problem showing two iterations of Newton's method. (b) Runtime needed by our SPGL1 algorithm to solve (LASSO$_\lambda$) for 512×4096 matrix A. Other lines show runtimes of other algorithms for solving the equivalent γ-weighted least-squares problem (2). (c)–(d) Runtimes needed by three algorithms for (BP$_\sigma$) with (c) $\sigma = 0$ and (d) $\sigma = 10^{-2}$. The horizontal axes of both show the number of rows m in a given problem, with $n = 4m$.

[3] E. G. Birgin, J. M. Martínez, and M. Raydan. Nonmonotone spectral projected gradient methods on convex sets. *SIAM J. Optim.*, 10(4):1196–1211, 2000.

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

[5] S. S. Chen, D. L. Donoho, and M. A. Saunders. Atomic decomposition by basis pursuit. *SIAM Rev.*, 43 (1):129–159, 2001.

[6] Y.-H. Dai and R. Fletcher. Projected Barzilai-Borwein methods for large-scale box-constrained quadratic programming. *Numer. Math.*, 100:21–47, 2005.

[7] I. Daubechies, M. Defrise, and C. D. Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Comm. Pure Appl. Math.*, 57:1413–1457, 2004.

[8] D. L. Donoho. For most large underdetermined systems of linear equations the minimal ℓ-norm solution is also the sparsest solution. *Comm. Pure Appl. Math.*, 59(6):797–829, 2006. 797 - 829.

[9] D. L. Donoho and Y. Tsaig. Fast solution of l1-norm minimization problems when the solution may be sparse. `http://www.stanford.edu/~tsaig/research.html`, October 2006.

[10] D. L. Donoho, I. Driori, V. C. Stodden, and Y. Tsaig. Sparselab. `http://sparselab.stanford.edu/`, 2007.

[11] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *Ann. Statist.*, 32(2):407–499, 2004.

[12] M. Figueiredo, R. Nowak, and S. J. Wright. Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems. `www.lx.it.pt/~mtf/GPSR`, February 2007.

[13] MOSEK. Mathematical programming system, `http://www.mosek.com`, 2007.

[14] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, New York, second edition, 2006.

[15] M. R. Osborne, B. Presnell, and B. A. Turlach. A new approach to variable selection in least squares problems. *IMA J. Numer. Anal.*, 20(3):389–403, 2000.

[16] R. T. Rockafellar. *Convex Analysis*. Princeton University Press, Princeton, 1970.

[17] J. F. Sturm. Using SeDuMi 1.02, a Matlab toolbox for optimization over symmetric cones (updated for Version 1.05). Technical report, Department of Econometrics, Tilburg University, Tilburg, The Netherlands, August 1998 – October 2001.

[18] J. A. Tropp. Just relax: Convex programming methods for identifying sparse signals in noise. *IEEE Trans. Info. Theory*, 52(3):1030–1051, March 2006.