

A FIRST-ORDER SMOOTHED PENALTY METHOD FOR COMPRESSED SENSING

N. S. AYBAT* AND G. IYENGAR†

Abstract. We propose a first-order smoothed penalty algorithm (SPA) to solve the sparse recovery problem $\min\{\|x\|_1 : Ax = b\}$. SPA is efficient as long as the matrix-vector product Ax and $A^T y$ can be computed efficiently; in particular, A need not be an orthogonal projection matrix. SPA converges to the target signal by solving a sequence of penalized optimization sub-problems, and each sub-problem is solved using Nesterov’s optimal algorithm for simple sets [13, 14]. We show that the SPA iterates x_k are ϵ -feasible, i.e. $\|Ax_k - b\|_2 \leq \epsilon$ and ϵ -optimal, i.e. $\|x_k\|_1 - \|x^*\|_1 \leq \epsilon$ after $\tilde{\mathcal{O}}(\epsilon^{-\frac{3}{2}})$ iterations. We also bound the sub-optimality, $\|x_k\|_1 - \|x^*\|_1$ for *any* iterate x_k ; thus, the user can stop the algorithm at any iteration k with guarantee on the sub-optimality. SPA is able to work with ℓ_1 , ℓ_2 or ℓ_∞ penalty on the infeasibility, and SPA can be easily extended to solve the relaxed recovery problem $\min\{\|x\|_1 : \|Ax - b\|_2 \leq \epsilon\}$

1. Introduction. In this paper we are interested in computing sparse solutions for the system of equations

$$Ax = b,$$

where $b \in \mathbb{R}^m$, $x \in \mathbb{R}^n$, $A \in \mathbb{R}^{m \times n}$ and the number of equations $m \ll n$. Define ℓ_0 -norm $\|x\|_0$ of the vector x as

$$\|x\|_0 = \sum_{i=1}^n \mathbf{1}(x(i) \neq 0),$$

where $x(i)$ denotes the i -th component of the vector x , and the indicator function $\mathbf{1}(x(i) \neq 0)$ takes the value 1 if $x(i) \neq 0$, and 0 otherwise. Then the sparsest solution x satisfying the equation $Ax = b$ can be recovered by solving the following ℓ_0 -minimization problem

$$\min_{x \in \mathbb{R}^n} \|x\|_0 \text{ subject to } Ax = b. \tag{1.1}$$

Unfortunately this optimization problem is NP-hard and is often hard to solve in practice. It was known that optimal solution to (1.1) can often be recovered by solving the ℓ_1 -minimization problem

$$\min_{x \in \mathbb{R}^n} \|x\|_1 \text{ subject to } Ax = b, \tag{1.2}$$

where $\|x\|_1 = \sum_{i=1}^n |x(i)|$ denotes ℓ_1 -norm. This problem can be reformulated into a linear program and therefore, can, in theory, be solved efficiently. Recently, Candes, Romberg and Tao [2, 3, 4] and Donoho [6] have shown that when the optimal solution x^* of (1.1) is s -sparse, i.e. only s of the n components are non-zeros, and the matrix A satisfies some regularity conditions, in particular $m = \mathcal{O}(s \ln(n))$, the sparsest solution x^* is also optimal for (1.2). Thus, the sparsest solution x^* can be recovered by solving a linear program (LP). This result has given rise to a new signal compression methodology known as *compressed sensing* (CS). In the CS methodology a high dimensional s -sparse signal x is compressed into $m = \mathcal{O}(s \ln(n))$ dimensional code $b = Ax$ by taking m linear measurements. When the measurement matrix A satisfies appropriate regularity conditions, the original signal x can be decoded by solving an LP.

Previous results. In practice, solving the decoding LP (1.2) is hard. This is because the constraint matrix A is large and dense, and the LPs are often ill-conditioned. Thus, general purpose LP solvers are not able to efficiently solve the CS LPs. On the other hand, the measurement matrix A often has a lot of structure. For example, A is often a partial Fourier matrix, i.e. $b = Ax$ is the discrete Fourier transform over a small set of frequencies. Consequently, Ax and $A^T y$ can be computed very efficiently using the Fast

*IEOR Department, Columbia University. Email: nsa2106@columbia.edu

†IEOR Department, Columbia University. Email: gi10@columbia.edu

Fourier Transform (FFT). Algorithms that are able to exploit this structure are likely to be very efficient for solving (1.2). A number of different research groups have proposed algorithms that exploit this structure. Some groups proposed solving ℓ_1 minimization in Lagrangian form:

$$\min_{x \in \mathbb{R}^n} \gamma \|x\|_1 + \|Ax - b\|_2^2. \quad (1.3)$$

Figueiredo, Nowak and Wright [12] propose the GPSR algorithm that uses gradient projection method with Barzila Borwein steps to solve (1.3). The algorithm proceeds by computing an approximately optimal solution $x^*(\gamma_k)$ for $\gamma = \gamma_k$ starting from the approximately optimal solution $x^*(\gamma_{k-1})$ for the previous value of γ .

Hale, Yin and Zhang [7, 8] propose a fixed point continuation (FPC) algorithm to solve (1.3). They use operator-splitting to show that the optimal solution of (1.3) satisfies a fixed-point equation $x^*(\gamma) = F(x^*(\gamma))$. They show that the operator F , which is a composition of gradient descent step operator and shrinkage operator, is a contraction mapping; therefore, the mapping $x_{k+1} = F(x_k)$ converges to the optimal solution. Wen, Yin, Goldfarb and Zhang [17] improve the performance of FPC by adding an active set (AS) optimization step. In the FPC-AS algorithm, once the shrinkage iterations produce a ‘‘candidate’’ solution x_k , the non-smooth objective function $\|x\|_1$ is replaced by $\text{sign}(x_k)^T x$ and the constraints $\text{sign}(x_k)(i)x(i) \geq 0$ are added. Since the objective function on the active set is smooth, one can use conjugate gradients or quasi-Newton methods to minimize the objective.

Yin, Osher, Goldfarb and Darbon [16] propose a method based on Bregman iterative regularization. The algorithm solves problem of the form

$$\min_{x \in \mathbb{R}^n} \gamma \|x\|_1 + \frac{1}{2} \|Ax - f_k\|_2^2, \quad (1.4)$$

where f_k are obtained by suitably updating the measurement vector b . This method utilizes FPC for solving unconstrained subproblems. Typically, only a few outer iterations are done, and for each outer iteration FPC [8] is called to solve subproblem (1.4).

Alternative algorithms for the unconstrained ℓ_1 problem include an iterative solver in an interior-point framework [11], and an accelerated projection gradient method [9]. Moreover, in [5], Van den Berg and Friedlander adapted the nonmonotone spectral projected gradient algorithm to solve the LASSO problem

$$\min_{\{x: \|x\|_1 \leq t\}} \|Ax - b\|_2^2.$$

New results. In this paper we propose a new penalty approach that solves the CS decoding problem (1.2) by solving a sequence of problems that are smoothed versions of the optimization problem

$$\min \{ \lambda \|x\|_1 + \|Ax - b\|_2 \}$$

Note that we penalize the infeasibility by $\|Ax - b\|_2$ and *not* by $\|Ax - b\|_2^2$. Since $\|Ax - b\|_2$ is known to be an *exact* penalty function, we expect our proposed method to have stronger convergence properties. Our method is an extension of the Nesterov’s non-smooth optimization algorithm [14]. The main results of this paper are as follows.

- (a) We show that our algorithm converges to an optimal solution x^* to (1.2), i.e. $x^* \in \text{argmin}\{\|x\|_1 : Ax = b\}$. See Theorem 2.1 and Corollary 2.2 for details. In order for the algorithm to be efficient, we only require that the matrix-vector product Ax and $A^T y$ be computed efficiently; in particular, we do not require that A be an orthogonal projection matrix. This implies that our algorithm can be used to recover compressed CT scans [1] where the measurement matrix A is *not* an orthogonal projection. Note that even in the special case when the measurement matrix is an orthogonal projection, the standard Nesterov algorithm for non-smooth minimization can only compute an ϵ -optimal solution for (1.2); it does not converge to an optimal solution.

- (b) We also establish a convergence rate for the algorithm. We show that there exist a priori fixed parameter settings such that, for *all* small enough ϵ , the iterates x_k computed by our algorithm are ϵ -feasible, i.e. $\|Ax_k - b\|_2 \leq \epsilon$, and ϵ -optimal, $\|x_k\|_1 - \|x^*\|_1 \leq \epsilon$, after $\tilde{O}(\epsilon^{-\frac{3}{2}})$ iterations. See Theorem 2.5 for details. To the best of our knowledge this is the first penalty based method for compressed sensing with a provable convergence rate.

Note that our algorithm does *not* require the user to tune the parameter setting for the chosen degree of approximation ϵ – the iterates x_k will be ϵ -optimal and ϵ -feasible for *any* small enough ϵ after $\tilde{O}(\epsilon^{-\frac{3}{2}})$ iterations. In contrast, the Nesterov algorithm for non-smooth optimization requires the user to specify ϵ a priori and computes an ϵ -optimal and ϵ -feasible solution to (1.2) in $\tilde{O}(\epsilon^{-1})$ iterations – iterating further does *not* improve the accuracy and one does not converge to a solution to (1.2). Thus, for any given ϵ the Nesterov algorithm has a better complexity bound; however, the user does not have the flexibility of iterating further to obtain a more accurate solution.

- (c) We also bound the sub-optimality, $\|x_k\|_1 - \|x^*\|_1$ for *any* iterate x_k . Thus, the user can stop the algorithm at any iteration k with guarantee on the sub-optimality. See Theorem 2.4 for details.
- (d) Our proposed algorithmic framework allows one also to use ℓ_1 or ℓ_∞ norm to penalize infeasibility. The framework easily extends to the relaxed recovery problem $\min\{\|x\|_1 : \|Ax - b\|_2 \leq \epsilon\}$.

The rest of the paper is organized as follows. In Section 2 we develop our proposed algorithm for solving the ℓ_1 -minimization problem. In Section 3 we discuss extensions of the algorithm to the related optimization problems. In Section 4 we discuss results of our numerical experiments.

2. A smoothed penalty method for ℓ_1 -minimization. In this section, we describe a penalty-function based method for solving (1.2). Let

$$\mathcal{X} = \{x \in \mathbb{R}^n | Ax = b\}$$

denote the feasible region of the problem (1.2). We will assume that A has full row rank. Consequently, A^T has full column rank. Let x^* denote an optimal solution of the ℓ_1 -minimization problem (1.2).

Let $P(x) = \|Ax - b\|_2$. Since $P(x)$ is an *exact* penalty function for \mathcal{X} , it follows that there exists a constant $\lambda^* > 0$ such that x^* is optimal for

$$\min \{\lambda \|x\|_1 + P(x)\} \quad \forall \lambda \leq \lambda^*. \quad (2.1)$$

The penalized optimization problem (2.1) is a convex optimization problem. However, both $\|x\|_1$ and $P(x)$ are non-smooth functions of x ; consequently, (2.1) is a non-smooth optimization problem and sub-gradient based methods are likely to perform rather poorly. In this section we propose an algorithm that solves an appropriately “smoothed” version of (2.1). The smoothing and the algorithm builds on the work of Nesterov [14].

Since $\|x\|_1 = \max_{\{u: \|u\|_\infty \leq 1\}} \{u^T x\}$, we smooth $\|x\|_1$ by setting

$$f_\mu(x) = \max_{\{u: \|u\|_\infty \leq 1\}} \left\{ x^T u - \frac{\mu}{2} (u^T u) \right\}$$

for $\mu > 0$. The function $f_\mu(x)$ is convex and continuously differentiable with

$$\nabla f_\mu(x) = u_x,$$

where

$$u_x(i) = \text{sign}(x(i)) \min \left\{ \frac{|x(i)|}{\mu}, 1 \right\}, \quad i = 1, \dots, n, \quad (2.2)$$

where

$$\text{sign}(x) = \begin{cases} 1 & x > 0, \\ 0 & x = 0, \\ -1 & x < 0. \end{cases}$$

The gradient $\nabla f_\mu(x)$ is Lipschitz continuous with the Lipschitz constant

$$L_\mu^f = \frac{1}{\mu}.$$

See [14] for details.

We smooth the penalty function $P(x)$ by setting

$$P_\nu(x) = \max_{\{w: \|w\|_2 \leq 1\}} \left\{ (Ax - b)^T w - \frac{\nu}{2} (w^T w) \right\}$$

for $\nu > 0$. Solving the optimization problem we get

$$P_\nu(x) = \begin{cases} \frac{\|Ax - b\|_2^2}{2\nu}, & \text{if } \|Ax - b\|_2 \leq \nu; \\ \|Ax - b\|_2 - \frac{\nu}{2}, & \text{if } \|Ax - b\|_2 > \nu. \end{cases} \quad (2.3)$$

The function $P_\nu(x)$ is convex, and continuously differentiable for all $x \in \mathbb{R}^n$ with

$$\nabla P_\nu(x) = A^T w_x,$$

where

$$w_x = \begin{cases} \frac{Ax - b}{\nu}, & \|Ax - b\|_2 \leq \nu; \\ \frac{Ax - b}{\|Ax - b\|_2}, & \|Ax - b\|_2 > \nu. \end{cases} \quad (2.4)$$

The gradient $\nabla P_\nu(x)$ is Lipschitz continuous with the Lipschitz constant

$$L_\nu^P = \frac{\|A\|_2^2}{\nu},$$

where $\|A\|_2 = \max_{\{u, v: \|u\|_2 \leq 1, \|v\|_2 \leq 1\}} \{u^T Av\} = \sigma_{\max}(A)$, where $\sigma_{\max}(A)$ denotes the largest singular value of A . Note that $P_\nu(x)$ is the Hübner penalty function that is often used in robust statistics.

We propose to solve (2.1) (or, equivalently, (1.2)) by solving a sequence of optimization problems of the form

$$\min_{x \in \mathbb{R}^n} \left\{ \lambda f_\mu(x) + P_\nu(x) \right\}. \quad (2.5)$$

The Smoothed Penalty Algorithm (SPA) that we use to solve (1.2) is displayed in Figure 2.1.

```

SMOOTHED PENALTY ALGORITHM ( $\{(\mu_k, \nu_k, \lambda_k, \tau_k)\}_{k \in \mathbb{Z}_+}$ )
   $q_0 \leftarrow \arg \min \{\|x\|_2 \mid Ax = b\}$ ,    $x_0 \leftarrow q_0$ ,    $\beta_1 = f_{\mu_1}(q_0)$ ,    $k \leftarrow 0$ 
  while (Stopping Criterion not true)
    do
       $k \leftarrow k + 1$ 
      Define  $Q_k(x) = \lambda_k f_{\mu_k}(x) + P_{\nu_k}(x)$  and  $\sigma_k \leftarrow \beta_k + \frac{\mu_k n}{2}$ 
      Starting from  $x_{k-1}$  compute  $x_k$  such that  $\|\nabla Q_k(x_k)\|_2 \leq \tau_k$  and  $\|x_k\|_1 \leq \sigma_k$ 
       $q_k \leftarrow \arg \min \{\|x - x_k\|_2 : Ax = b\}$ 
       $\beta_{k+1} \leftarrow \min_{0 \leq j \leq k} \{f_{\mu_{k+1}}(q_j)\}$ 
  return  $x_k$ 

```

FIG. 2.1. Smoothed Penalty Algorithm (SPA)

THEOREM 2.1. Let $\{x_k \in \mathbb{R}^n : k \in \mathbb{Z}_+\}$ denote the sequence of iterates generated by the Smoothed Penalty Algorithm (SPA) displayed in Figure 2.1 when

- (i) Smoothing parameter for $\|x\|_1$: $\mu_k \searrow 0$
- (ii) Smoothing parameter for $P(x)$: $\nu_k \searrow \alpha \geq 0$
- (iii) Penalty multiplier: $\lambda_k \searrow 0$
- (iv) Approximate optimality parameter: $\tau_k \searrow 0$ such that $\frac{\tau_k}{\lambda_k} \rightarrow 0$.

Then, $\{x_k \in \mathbb{R}^n : k \in \mathbb{Z}_+\}$ is a bounded sequence. Let \bar{x} denote any limit point of $\{x_k : k \in \mathbb{Z}_+\}$. Then \bar{x} is an optimal solution of the ℓ_1 -minimization problem (1.2).

Remark 2.1. The notation $\gamma_k \searrow \eta$ (resp. $\gamma_k \nearrow \eta$) denotes that the sequence $\{\gamma_k\}$ is monotonically decreasing (resp. increasing).

Proof. Since $\mu_k \searrow 0$ and Step 2 of SPA implies that $\beta_k \leq f_{\mu_k}(q_0) \leq \|q_0\|_1$, the iterates $\{x_k\}_{k \in \mathbb{Z}_+}$ lie in a bounded set $\|x\|_1 \leq \|q_0\|_1 + \frac{\mu_1 n}{2}$. Let \bar{x} denote any limit point of this sequence and let \mathcal{K} denote a subsequence such that $\lim_{k \in \mathcal{K}} x_k = \bar{x}$.

The gradient $\nabla Q_k(x_k) = u_k + A^T w_k$ where u_k satisfies (2.2) with $x = x_k$ and $\mu = \mu_k$, and w_k satisfies (2.4) with $x = x_k$ and $\nu = \nu_k$. Step 1 implies that

$$\|\nabla Q_k(x_k)\|_2 = \|\lambda_k u_k + A^T w_k\|_2 \leq \tau_k.$$

Thus, it follows that

$$\|A^T w_k\|_2 \leq (\tau_k + \lambda_k \|u_k\|_2) \leq (\tau_k + \lambda_k \sqrt{n}),$$

where the last inequality follows from the fact $\|u_k\|_2 \leq \sqrt{n}$. Hence,

$$\lim_{k \in \mathcal{K}} \|A^T w_k\|_2 = 0.$$

Since $\|\cdot\|_2$ is a continuous function, and A^T is assumed to have a full column rank, it follows that

$$\lim_{k \in \mathcal{K}} \|A^T w_k\|_2 = 0 \Rightarrow \|A^T \lim_{k \in \mathcal{K}} w_k\|_2 = 0 \Rightarrow \lim_{k \in \mathcal{K}} w_k = 0.$$

Thus, $\exists B > 0$ such that $\|w_k\|_2 < 1$ for $k \geq \mathcal{K} \cap \{l : l \geq B\}$. From equation (2.4), it follows that for all $k \in \mathcal{K} \cap \{l : l \geq B\}$

$$w_k = \frac{1}{\nu_k} (Ax_k - b).$$

Consequently, $\lim_{k \in \mathcal{K}} w_k = 0$ implies that the limit point \bar{x} satisfies

$$A\bar{x} = b \tag{2.6}$$

Note that this proof works for any $\alpha \geq 0$, i.e. we don't need to force the penalty parameter $\nu_k \rightarrow 0$.

Since $\|\nabla f_{\mu_k}(x_k)\|_\infty = \|u_k\|_\infty \leq 1$ for all $k \in \mathbb{Z}_+$, there exists a vector $\bar{g} \in \mathbb{R}^n$ and a subsequence $\mathcal{K}_1 \subset \mathcal{K}$ such that

$$\lim_{k \in \mathcal{K}_1} u_k = \bar{g}. \tag{2.7}$$

Since $\lim_{k \in \mathcal{K}_1} x_k = \bar{x}$ and (2.7) hold, it follows that

$$\bar{g}(i) = \begin{cases} \text{sign}(\bar{x}(i)) & |\bar{x}(i)| \neq 0, \\ \in [-1, 1] & \bar{x}(i) = 0. \end{cases}$$

Thus,

$$\bar{g} \in \partial \|x\|_1 |_{x=\bar{x}}. \tag{2.8}$$

Let $\theta_k = \frac{-w_k}{\lambda_k}$. Since the gradient $\nabla Q_k(x) = \lambda_k u_k + A^T w_k$ we have that

$$A^T \theta_k = u_k - \frac{1}{\lambda_k} \nabla Q_k(x_k). \quad (2.9)$$

Since A^T has full column rank, we have that

$$\theta_k = (AA^T)^{-1} A(u_k - \frac{1}{\lambda_k} \nabla Q_k(x_k)).$$

From Step 1 we have that $\|\nabla Q_k(x_k)\|_2 \leq \tau_k$ and $\frac{\tau_k}{\lambda_k} \rightarrow 0$; therefore, $\bar{\theta} = \lim_{k \in \mathcal{K}_1} \theta_k$ exists. Hence, (2.9) implies that

$$\bar{g} = A^T \bar{\theta}. \quad (2.10)$$

From (2.6)-(2.10), it follows that \bar{x} is a Karush-Kuhn-Tucker (KKT) point for the ℓ_1 -minimization problem (1.2). Since $\|x\|_1$ is convex, the optimization problem (1.2) is a convex programming problem. Hence KKT conditions are sufficient for optimality and we can conclude that \bar{x} is an optimal solution for (1.2). \square

In compressed sensing exact recovery occurs only when $\min\{\|x\|_1 : Ax = b\}$ has a *unique* solution. The following Corollary establishes that SPA converges to this solution.

COROLLARY 2.2. *Suppose the ℓ_1 -minimization problem $\min\{\|x\|_1 : Ax = b\}$ has a unique optimal solution. Let $\{x_k : k \in \mathbb{Z}_+\}$ denote the sequence of iterates generated by the Smoothed Penalty Algorithm (SPA) displayed in Figure 2.1 when*

- (i) *Smoothing parameter for $\|x\|_1$: $\mu_k \searrow 0$*
- (ii) *Smoothing parameter for $P(x)$: $\nu_k \searrow \alpha \geq 0$*
- (iii) *Penalty multiplier: $\lambda_k \searrow 0$*
- (iv) *Approximate optimality parameter: $\tau_k \searrow 0$ such that $\frac{\tau_k}{\lambda_k} \rightarrow 0$.*

Then $\lim_{k \rightarrow \infty} x_k = x^$ where $x^* = \arg \min\{\|x\|_1 : Ax = b\}$.*

Proof. From Theorem 2.1 we have that every limit point \bar{x} of the sequence of iterates $\{x_k : k \in \mathbb{Z}_+\}$ generated by SPA is an optimal solution of the ℓ_1 -minimization problem (1.2). Since the optimal solution is unique, it follows that the bounded sequence $\{x_k : k \in \mathbb{Z}_+\}$ has a unique limit point. Consequently, the sequence has a limit and the limit is the optimal solution of (1.2). \square

In order to complete the analysis, we need to establish that Step 1 can always be satisfied, i.e. for all $k \in \mathbb{Z}^+$ there exists x_k such that $\|\nabla Q^{(k)}(x_k)\|_2 \leq \tau_k$ and $\|x_k\|_1 \leq \sigma_k$.

LEMMA 2.3. *Nesterov's optimal algorithm for simple sets [13] computes an x_k satisfying Step 1 in SPA in*

$$N_k = \left\lceil \frac{4L_k \sigma_k}{\tau_k} \right\rceil \quad (2.11)$$

iterations, where $L_k = \frac{\lambda_k}{\mu_k} + \frac{\|A\|_2^2}{\nu_k}$ denotes the Lipschitz constant for ∇Q_k .

Remark 2.2. *Note that Nesterov's optimal algorithm for simple sets guarantees the bound (2.11) for all initial starting points for the k -th subproblem. We are not able to take advantage of the fact that the particular initial point $x_{(k-1)}$ for the k -th subproblem is close to an optimal solution for the k -th subproblem since $Q_{k-1}(x) \approx Q_k(x)$ for all x .*

Proof. Let $\{x_k\}$ denote the iterates computed by SPA. Then SPA define

$$\begin{aligned} q_0 &= \operatorname{argmin}\{\|x\|_2 : Ax = b\}, & q_k &= \operatorname{argmin}\{\|x - x_k\|_2 : Ax = b\}, \\ \beta_1 &= f_{\mu_1}(q_0), & \beta_{k+1} &= \min_{0 \leq j \leq k} \{f_{\mu_{k+1}}(q_j)\}, \end{aligned}$$

and $\sigma_k = \beta_k + \frac{\mu_k \tau_k}{2}$.

Let $x_k^* = \operatorname{argmin}_{x \in \mathbb{R}^n} Q^{(k)}(x)$ denote the unconstrained optimal solution. Since $P_\mu(q_j) = 0$ for all $\mu > 0$ and $j \geq 1$, it follows that

$$Q_k(x_k^*) = \lambda_k f_{\mu_k}(x_k^*) + P_k(x_k^*) \leq \lambda_k \min_{0 \leq j \leq k-1} \{f_{\mu_k}(q_j)\} = \lambda_k \beta_k.$$

The smoothed function $f_{\mu_k}(x) \geq \|x\|_1 - \frac{\mu_k n}{2}$ and $P_\nu(x) \geq 0$ for all $\mu > 0$, $\nu > 0$ and $x \in \mathbb{R}^n$. Therefore,

$$\|x_k^*\|_1 - \frac{\mu_k n}{2} \leq f_{\mu_k}(x_k^*) \leq \lambda_k^{-1} Q_k(x_k^*) \leq \beta_k.$$

Thus,

$$\|x_k^*\|_1 \leq \beta_k + \frac{\mu_k n}{2} \triangleq \sigma_k.$$

Thus, $\min_{x \in \mathbb{R}^n} Q_k(x)$ is equivalent to $\min\{Q_k(x) \mid \|x\|_1 \leq \sigma_k\}$.

Let $g : \mathbb{R}^n \rightarrow \mathbb{R}$ be a convex function and have a Lipschitz continuous gradient with constant L with respect to the ℓ_2 norm. Let $x^* = \operatorname{argmin}\{g(x) : \|x\|_1 \leq \sigma\}$. Then, after l iterations of Nesterov's optimal algorithm for simple sets [13, 14] it is guaranteed that

$$g(x_l) - g(x^*) \leq \frac{8L\beta^2}{(l+1)(l+2)}.$$

For any convex function with a Lipschitz continuous gradient, we have that

$$\frac{1}{2L} \|\nabla g(x_l)\|_2^2 \leq g(x_l) - g(x^*).$$

Thus, $\|\nabla g(x_l)\|_2 \leq \tau$ for all $l \geq \lceil \frac{4L\sigma}{\tau} \rceil$.

Since $Q_k(x)$ is a convex function with a Lipschitz continuous gradient with Lipschitz constant $L_k = \frac{\lambda_k}{\mu_k} + \frac{\|A\|_2^2}{\nu_k}$, Nesterov's optimum algorithm guarantees that

$$\|\nabla Q^{(k)}(x_l)\|_2 \leq \tau_k$$

for all $l \geq \lceil \frac{4L_k \sigma_k}{\tau_k} \rceil \triangleq N_k$. \square

The SPA with all the details is shown in Figure 2.2. The details of Nesterov's optimal algorithm for simple sets [13, 14] for computing

$$\begin{aligned} & \text{minimize} && g(x), \\ & \text{subject to} && \|x\|_1 \leq \sigma, \end{aligned}$$

where g is a convex function with a Lipschitz continuous gradient with constant L are as follows. The algorithm computes three sets of iterates (x_l, y_l, z_l) :

1. y_l is computed using x_l and the gradient $\nabla g(x_l)$:

$$\begin{aligned} y_l &= \operatorname{argmin} \left\{ \nabla g(x_l)^T y + \frac{L}{2} \|y - x_l\|_2^2 : \|y\|_1 \leq \sigma \right\}, \\ &= \operatorname{argmin} \left\{ \left\| y - \left(x_l - \frac{1}{L} \nabla g(x_l) \right) \right\|_2^2 : \|y\|_1 \leq \sigma \right\}, \\ &= \Pi_{\ell_1} \left(\sigma, x_l - \frac{1}{L} \nabla g(x_l) \right), \end{aligned}$$

where the function

$$\Pi_{\ell_1}(\sigma, \hat{y}) = \operatorname{argmin}\{\|y - \hat{y}\|_2^2 : \|y\|_1 \leq \sigma\}.$$

Thus, in Step 1 in Figure 2.2 we compute the y_l iterate. In Appendix A.3 we show that the projection Π_{ℓ_1} can be computed efficiently.

2. z_l is computed using the initial iterate x_0 and the gradients $\nabla g(x_i)$ for all the iterates $i \leq l$:

$$\begin{aligned} z_l &= \operatorname{argmin} \left\{ \sum_{i=0}^l \left(\frac{i+1}{2} \right) \nabla g(x_i)^T z + \frac{L}{2} \|z - x_0\|_2^2 : \|z\|_1 \leq \sigma \right\}, \\ &= \operatorname{argmin} \left\{ \left\| z - \left(x_l - \frac{1}{L} \sum_{i=0}^l \left(\frac{i+1}{2} \right) \nabla g(x_i) \right) \right\|_2^2 : \|z\|_1 \leq \sigma \right\}, \\ &= \Pi_{\ell_1} \left(\sigma, x_0 - \frac{1}{L} \sum_{i=0}^l \left(\frac{i+1}{2} \right) \nabla g(x_i) \right). \end{aligned}$$

Thus, in Step 2 we compute the z_l iterate.

3. x_{l+1} is a convex combination of y_l and z_l :

$$x_{l+1} = \left(\frac{2}{l+3} \right) z_l + \left(\frac{l+1}{l+3} \right) y_l.$$

The SPA displayed in Figure 2.2 does not explicitly state a termination condition. In Section 4.2 we discuss the specific termination conditions used in our numerical experiments.

SMOOTHED PENALTY ALGORITHM $(\{(\mu_k, \nu_k, \lambda_k, \tau_k)\}_{k \in \mathbb{Z}_+})$

```

 $q_0 \leftarrow \operatorname{argmin}\{\|x\|_2 \mid Ax = b\}, \quad x_0 \leftarrow q_0, \quad \beta_1 = f_{\mu_1}(q_0), \quad k \leftarrow 0$ 
while (Stopping Criterion not true)
  do
     $k \leftarrow k + 1$ 
    while (Stopping Criterion not true)
      do
         $L_k \leftarrow \frac{\lambda_k}{\mu_k} + \frac{\|A\|_2^2}{\nu_k}, \quad x_{k,0} \leftarrow x_{(k-1)}, \quad \sigma_k \leftarrow \beta_k + \frac{\mu_k n}{2}$ 
         $l \leftarrow 0$ 
        compute  $\nabla Q_k(x_{k,l})$ 
        while  $(\|\nabla Q_k(x_{k,l})\|_2 > \tau_k)$ 
          do
            1  $y_{k,l} \leftarrow \Pi_{\ell_1} \left( \sigma_k, x_{k,l} - \frac{\nabla Q_k(x_{k,l})}{L_k} \right)$ 
            2  $z_{k,l} \leftarrow \Pi_{\ell_1} \left( \sigma_k, x_{k,0} - \frac{\sum_{i=0}^l \left( \frac{i+1}{2} \right) \nabla Q_k(x_{k,i})}{L_k} \right)$ 
             $x_{k,l+1} \leftarrow \left( \frac{2}{l+3} \right) z_{k,l} + \left( \frac{l+1}{l+3} \right) y_{k,l}$ 
             $l \leftarrow l + 1$ 
          compute  $\nabla Q_k(x_{k,l})$ 

         $x_k \leftarrow x_{k,l}$ 
         $q_k \leftarrow \operatorname{argmin}\{\|x - x_k\|_2 : Ax = b\}$ 
         $\beta_{k+1} \leftarrow \min_{1 \leq j \leq k} \{f_{\mu_{k+1}}(q_j)\}$ 
         $k \leftarrow k + 1$ 

return  $x_k$ 

```

FIG. 2.2. Details of the Smoothed Penalty Algorithm (SPA)

Next, we prove a finite iteration result for SPA.

THEOREM 2.4. *Let $\{x_k\}_{k \in \mathbb{Z}_+}$ denote the sequence of iterates generated by the SPA displayed in Figure 2.2. Then, for all $k \geq 1$,*

$$\|x_k\|_1 \leq \|x^*\|_1 + \frac{\tau_k^2}{2\lambda_k L_k} + \frac{\mu_k n}{2}.$$

Let $\sigma_{\min}(A)$ denote the smallest non-zero singular value of A . Then, for all k such that $\tau_k + \lambda_k \sqrt{n} < \sigma_{\min}(A)$,

$$\|Ax_k - b\|_2 \leq \nu_k, \quad \text{and} \quad \|x_k\|_1 \geq \|x^*\|_1 - \frac{\mu_k n}{2} - \frac{\nu_k}{\lambda_k}.$$

Proof. Since Q_k has a Lipschitz continuous gradient with the Lipschitz constant L_k , $\|\nabla Q_k(x_k)\|_2 \leq \tau_k$ implies that

$$Q_k(x_k) \leq Q_k(x_k^*) + \frac{\tau_k^2}{2L_k},$$

where $x_k^* = \operatorname{argmin}_x Q_k(x)$. Since the penalty function $P_\nu(x) \geq 0$, $f_\mu(x) \geq \|x\|_1 - \frac{\mu n}{2}$ for all $x \in \mathbb{R}^n$ and $\mu, \nu > 0$, it follows that

$$\begin{aligned} \lambda_k \|x_k\|_1 &\leq \lambda_k f_{\mu_k}(x_k) + \frac{\lambda_k \mu_k n}{2}, \\ &\leq Q_k(x_k) + \frac{\lambda_k \mu_k n}{2}, \\ &\leq Q_k(x_k^*) + \frac{\tau_k^2}{2L_k} + \frac{\lambda_k \mu_k n}{2}, \\ &\leq Q_k(x^*) + \frac{\tau_k^2}{2L_k} + \frac{\lambda_k \mu_k n}{2}, \\ &\leq \lambda_k \|x^*\|_1 + \frac{\tau_k^2}{2L_k} + \frac{\lambda_k \mu_k n}{2}, \end{aligned}$$

where the last inequality follows from the fact that $Ax^* = b$. Hence we have $\|x_k\|_1 \leq \|x^*\|_1 + \frac{\tau_k^2}{2\lambda_k L_k} + \frac{\mu_k n}{2}$.

Fix an iterate k such that $\tau_k + \lambda_k \sqrt{n} < \sigma_{\min}(A)$. Recall that $\|\nabla Q_k(x_k)\|_2 \leq \tau_k$ implies that

$$\|\nabla P_{\nu_k}(x_k)\|_2 = \|A^T w_k\|_2 \leq (\tau_k + \lambda_k \|\nabla f_{\mu_k}(x_k)\|_2) \leq \tau_k + \lambda_k \sqrt{n},$$

where the last inequality follows from the fact that $\|\nabla f_{\mu_k}(x_k)\|_\infty = \|u_k\|_\infty \leq 1$. Since A^T is assumed to have a full column rank, it follows that

$$\|w_k\|_2 \leq \frac{\|A^T w_k\|_2}{\sigma_{\min}(A)} \leq \frac{\tau_k + \lambda_k \sqrt{n}}{\sigma_{\min}(A)} < 1.$$

Since $\|w_k\| < 1$, (2.4) implies that

$$w_k = \frac{Ax_k - b}{\nu_k}.$$

Thus,

$$\|Ax_k - b\|_2 = \nu_k \|w_k\|_2 \leq \nu_k.$$

Next, we establish a lower bound for $\|x_k\|$ using the linear programming duality:

$$\begin{aligned} \text{minimize } \|x\|_1, & & = & & \text{maximize } b^T w, \\ \text{subject to } Ax = b. & & & & \text{subject to } \|A^T w\|_\infty \leq 1. \end{aligned}$$

Let w^* denote the optimal dual solution. Then $\|A^T w^*\|_2 \leq \sqrt{n}$ and $\|w^*\|_2 \leq \frac{\sqrt{n}}{\sigma_{\min}(A)}$.

Linear programming duality also implies that

$$\begin{aligned} \underset{x \in \mathfrak{R}^n}{\text{minimize}} \quad & \lambda \|x\|_1 + \|Ax - b\|_2 & = & \quad \text{maximize} \quad \lambda b^T w, \\ & & & \text{subject to} \quad \|A^T w\|_\infty \leq 1, \\ & & & \|w\|_2 \leq \lambda^{-1}. \end{aligned} \quad (2.12)$$

It is easy to check that w^* is feasible for the dual program in (2.12) whenever $\lambda\sqrt{n} \leq \sigma_{\min}(A)$.

Next, we relate exact penalty function to the smoothed penalty function $Q_k(x)$.

$$\begin{aligned} \min_x Q_k(x) & \geq \min_x \{ \lambda_k \|x\|_1 + \|Ax - b\|_2 \} - \frac{\lambda_k \mu_k n}{2} - \frac{\nu_k}{2}, \\ & \geq \lambda_k b^T w^* - \frac{\lambda_k \mu_k n}{2} - \frac{\nu_k}{2}, \\ & = \lambda_k \|x^*\|_1 - \frac{\lambda_k \mu_k n}{2} - \frac{\nu_k}{2}, \end{aligned} \quad (2.13)$$

where (2.13) holds for the iterate k since $\tau_k + \lambda_k \sqrt{n} < \sigma_{\min}(A)$ implies w^* is feasible for dual program in (2.12), and the inequality then follows from weak duality.

Finally,

$$\begin{aligned} Q_k(x_k) & = \lambda_k f_{\mu_k}(x_k) + P_{\nu_k}(x_k), \\ & \leq \lambda_k \|x_k\|_1 + \frac{\|Ax_k - b\|_2^2}{2\nu_k}, \end{aligned} \quad (2.14)$$

$$\leq \lambda_k \|x_k\|_1 + \frac{\nu_k}{2}, \quad (2.15)$$

where (2.14) and (2.15) follow from the fact that $\frac{\tau_k + \sqrt{n}\lambda_k}{\sigma_{\min}(A)} < 1$ implies that $w_k = \frac{Ax_k - b}{\nu_k}$ and $\|Ax_k - b\|_2 \leq \nu_k$. Thus,

$$\|x_k\|_1 \geq \|x^*\|_1 - \frac{\mu_k n}{2} - \frac{\nu_k}{\lambda_k}.$$

□

The proof of Theorem 2.1 and Theorem 2.4 reveals that the following relations must hold for the parameter set $(\mu, \nu, \lambda, \tau)$:

- (i) The penalty multiplier λ and the smoothing parameter for the penalty term ν must satisfy $\nu_k/\lambda_k \rightarrow 0$.
- (ii) The penalty multiplier λ and the approximate optimality parameter τ must satisfy $\tau_k/\lambda_k \rightarrow 0$.
- (iii) The penalty multiplier $\lambda = \mathcal{O}(\frac{1}{\sqrt{n}})$, the smoothing parameter for the ℓ_1 -norm $\mu = \mathcal{O}(\frac{1}{n})$, and the smoothing parameter for the penalty $\nu = \mathcal{O}(\lambda) = \mathcal{O}(\frac{1}{\sqrt{n}})$.

THEOREM 2.5. Fix $0 < \delta < 1$, $\frac{1}{4} \leq \alpha \leq \frac{3}{4}$, and strictly positive parameters $(\lambda_0, \tau_0, u_0, \nu_0)$. Then there exists a sequence of multipliers $\{(\lambda_k, \mu_k, \nu_k, \tau_k) : k \geq 1\}$ such that for all

$$0 < \epsilon < \frac{\max\{\mu_0, \frac{2\nu_0}{\lambda_0}, \frac{2\nu_0\tau_0^2}{\lambda_0\|A\|_2^2}, \frac{\nu_0}{\sqrt{n}}\} \sigma_{\min}(A)^{\frac{2+\delta}{\delta}} \alpha^{2+\delta}}{(2 \max\{\lambda_0, \tau_0\})^{\frac{2+\delta}{\delta}}}, \quad (2.16)$$

the SPA, displayed in Figure 2.2, computes a solution \bar{x} that is ϵ -feasible, i.e. $\|A\bar{x} - b\|_2 \leq \epsilon$, and ϵ -optimal, $\|\|\bar{x}\|_1 - \|x^*\|_1\| \leq \epsilon$, in $\mathcal{O}\left(\left(\|q_0\|_1 + \frac{\mu_0}{2}\right)n^{\frac{3}{2}} \ln(n) \epsilon^{-\frac{3}{2}(1+\delta)}\right)$ operations, where $q_0 = \operatorname{argmin}\{\|x\|_2 : Ax = b\}$.

Remark 2.3.

(a) A single fixed sequence $\{(\lambda_k, \mu_k, \nu_k, \tau_k) : k \geq 1\}$ of multipliers works for all sufficiently small ϵ .

(b) The bound (2.16) on the approximation ϵ is in practice not unreasonably small. For example, consider the case where target signal size $n = 512 \times 512 - 1$ with $s = \frac{n}{40}$ non-zero components and the measurements are the discrete Fourier transforms evaluated at $m = \frac{n}{8}$ randomly selected frequencies.

We show in Appendix A.1 that $\sigma_{\min}(A) = \frac{1}{\sqrt{2}}$ when the measurements are discrete Fourier transforms. In our numerical experiments (see Section 4) we chose

$$(\lambda_0, \tau_0, \mu_0, \nu_0) = (1, 0.2, 0.45\|q_0\|_1, 0.45\|q_0\|_1)$$

and the average value for $\|q_0\|_1$ over 10 randomly generated problem was $\|q_0\|_1 \approx 305707968.54166$. For $\delta = 0.125$ and $\alpha = 0.5$, the bound (2.16) implies that ϵ -optimal and ϵ -feasible solutions can be computed in $\mathcal{O}(\epsilon^{-1.6875})$ operations for all $\epsilon < 1.32$.

Proof. Set

$$\begin{aligned}\mu_1 &= \frac{\mu_0}{n}, \\ \nu_1 &= \frac{\nu_0}{\sqrt{n}}, \\ \tau_1 &= \tau_0, \\ \lambda_1 &= \frac{\lambda_0}{\sqrt{n}}.\end{aligned}$$

Choose $0 < \alpha < 1$ and update the parameters as follows: for all $k \geq 1$,

$$\begin{aligned}\tau_{k+1} &= \alpha^{\frac{1}{2}(1+\delta)} \cdot \tau_k, \\ \lambda_{k+1} &= \alpha^\delta \cdot \lambda_k, \\ \mu_{k+1} &= \alpha^{(1+\delta)} \cdot \mu_k, \\ \nu_{k+1} &= \alpha^{(1+\delta)} \cdot \nu_k.\end{aligned}\tag{2.17}$$

For the update scheme in (2.17),

$$\tau_k + \lambda_k \sqrt{n} \leq \tau_0 \alpha^{\frac{1}{2}(1+\delta)(k-1)} + \lambda_0 \alpha^{\delta(k-1)} \leq 2 \max\{\lambda_0, \tau_0\} \alpha^{\delta(k-1)}.$$

Thus, $\tau_k + \lambda_k \sqrt{n} \leq \sigma_{\min}(A)$ for all

$$k > K + 1 \triangleq \frac{\ln\left(\frac{2 \max\{\lambda_0, \tau_0\}}{\sigma_{\min}(A)}\right)}{\delta \ln\left(\frac{1}{\alpha}\right)} + 1.$$

Theorem 2.4 guarantees that for all $k > K + 1$,

$$\begin{aligned}\|x_k\|_1 - \|x^*\|_1 &\leq \frac{\mu_k n}{2} + \max\left\{\frac{\nu_k}{\lambda_k}, \frac{\tau_k^2}{2\lambda_k L_k}\right\}, \\ &\leq \frac{\mu_0}{2} \cdot \alpha^{(1+\delta)(k-1)} + \max\left\{\frac{\nu_0}{\lambda_0} \cdot \alpha^{(k-1)}, \frac{\nu_0 \tau_0^2}{\lambda_0 \|A\|_2^2} \cdot \alpha^{(2+\delta)(k-1)}\right\},\end{aligned}$$

where we have used the fact that $L_k \geq \frac{\|A\|_2^2}{\nu_k}$. Thus,

$$\|x_k\|_1 - \|x^*\|_1 \leq \epsilon,$$

for all

$$k > K_2 + 1 \triangleq \max\left\{K, \frac{\ln\left(\frac{\mu_0}{\epsilon}\right)}{(1+\delta) \ln\left(\frac{1}{\alpha}\right)}, \frac{\ln\left(\frac{2\nu_0}{\lambda_0 \epsilon}\right)}{\ln\left(\frac{1}{\alpha}\right)}, \frac{\ln\left(\frac{2\nu_0 \tau_0^2}{\lambda_0 \|A\|_2^2 \epsilon}\right)}{(2+\delta) \ln\left(\frac{1}{\alpha}\right)}\right\} + 1.\tag{2.18}$$

From Theorem 2.4 we also have that for $k > K + 1$,

$$\|Ax - b\|_2 \leq \nu_k = \frac{\nu_0}{\sqrt{n}} \alpha^{(1+\delta)(k-1)}.$$

Thus $\|Ax - b\|_2 \leq \epsilon$ for all

$$k > K_3 + 1 \triangleq \left\{ K, \frac{\ln\left(\frac{\nu_0}{\epsilon\sqrt{n}}\right)}{(1+\delta)\ln\left(\frac{1}{\alpha}\right)} \right\} + 1. \quad (2.19)$$

Let N_{out} denote the number of outer iterations required to compute an ϵ -infeasible and ϵ -optimal solution. Define

$$B = \max \left\{ \mu_0, \frac{2\nu_0}{\lambda_0}, \frac{2\nu_0\tau_0^2}{\lambda_0\|A\|_2^2}, \frac{\nu_0}{\sqrt{n}} \right\}.$$

Then (2.18) and (2.19) imply that for all $\epsilon < \frac{B\sigma_{\min}(A)^{\frac{2+\delta}{\delta}}\alpha^{2+\delta}}{(2\max\{\lambda_0, \tau_0\})^{\frac{2+\delta}{\delta}}}$,

$$N_{out} \leq \frac{\ln\left(\frac{B}{\epsilon}\right)}{\ln\left(\frac{1}{\alpha}\right)}. \quad (2.20)$$

Lemma 2.3 implies that N_{out} outer iterations of Algorithm SPA require a total of

$$N_{in} \leq \sum_{k=1}^{N_{out}} \left\lceil \frac{4L_k\sigma_k}{\tau_k} \right\rceil$$

inner iterations. The parameter $\sigma_k \leq \|q_0\|_1 + \frac{\mu_0}{2}$ for all $k \geq 1$, and

$$\begin{aligned} \sum_{k=1}^{N_{out}} \frac{L_k}{\tau_k} &= \sum_{k=1}^{N_{out}} \left(\frac{\lambda_k}{\tau_k\mu_k} + \frac{\|A\|_2^2}{\tau_k\nu_k} \right), \\ &= \sum_{k=0}^{N_{out}-1} \left(\left(\frac{\lambda_0\sqrt{n}}{\tau_0\mu_0} \right) \cdot \alpha^{-\frac{1}{2}(3+\delta)k} + \left(\frac{\|A\|_2^2\sqrt{n}}{\tau_0\nu_0} \right) \alpha^{-\frac{3}{2}(1+\delta)k} \right). \end{aligned}$$

Thus,

$$N_{in} = \mathcal{O} \left(\sqrt{n} \left(\|q_0\|_1 + \frac{\mu_0}{2} \right) \max \left\{ \frac{\lambda_0}{\tau_0\mu_0}, \frac{\|A\|_2^2}{\tau_0\nu_0} \right\} \alpha^{-\frac{3}{2}(1+\delta)N_{out}} \right).$$

From (2.20) it follows that for all $\epsilon < \frac{B\sigma_{\min}(A)^{\frac{2+\delta}{\delta}}\alpha^{2+\delta}}{(2\max\{\lambda_0, \tau_0\})^{\frac{2+\delta}{\delta}}}$,

$$N_{in} = \mathcal{O} \left(\sqrt{n} \left(\|q_0\|_1 + \frac{\mu_0}{2} \right) \max \left\{ \frac{\lambda_0}{\tau_0\mu_0}, \frac{\|A\|_2^2}{\tau_0\nu_0} \right\} \cdot B^{\frac{3}{2}(1+\delta)} \cdot \epsilon^{-\frac{3}{2}(1+\delta)} \right).$$

Since each inner iteration requires $\mathcal{O}(n \ln(n))$ operations, it follows that algorithm SPA computes an ϵ -infeasible and ϵ -optimal solution in $\mathcal{O} \left(\left(\|q_0\|_1 + \frac{\mu_0}{2} \right) n^{\frac{3}{2}} \ln(n) \epsilon^{-\frac{3}{2}(1+\delta)} \right)$ operations.

□

3. Extensions of the SPA.

3.1. The case of ℓ_1 and ℓ_∞ norms. We define the penalty for infeasibility in terms of the ℓ_2 norm. All the results in Section 2 hold when the penalty $P(x) = \|Ax - b\|_1$ or $P(x) = \|Ax - b\|_\infty$.

First consider the case where the penalty

$$P(x) = \|Ax - b\|_1 = \max_{\{w: \|w\|_\infty \leq 1\}} w^T (Ax - b).$$

In this case, the smoothed penalty function

$$P_\nu(x) = \max_{\{w: \|w\|_\infty \leq 1\}} \left\{ w^T (Ax - b) - \frac{\nu}{2} \|w\|_2^2 \right\},$$

has a Lipschitz continuous gradient

$$\nabla P_\nu(x) = A^T w_x$$

where

$$w_x(i) = \text{sign}(a_i^T x - b(i)) \min \left\{ \frac{a_i^T x - b(i)}{\nu}, 1 \right\}, \quad i = 1, \dots, m, \quad (3.1)$$

where a_i denotes the i -th row of A . Let $\{x_k\}_{k \geq 1}$ denote the sequence of iterates generated by SPA using the ℓ_1 penalty function. Note that changing the penalty $P_\nu(x)$ only changes the gradient $\nabla Q_k(x_k)$ – all the other steps remain the same. Let \bar{x} denote any limit point of the sequence $\{x_k\}_{k \geq 1}$, i.e. there exists a subsequence $\{x_k\}_{k \in \mathcal{K}}$ such that $\lim_{k \rightarrow \infty} x_k = \bar{x}$. We will show that \bar{x} is feasible, i.e. $A\bar{x} = b$.

The stopping condition $\|\nabla Q_k(x_k)\| \leq \tau_k$ and the fact that A^T has full column rank implies that $\lim_{k \in \mathcal{K}} w_k = 0$, i.e. there exists B such that for all $k \geq B$, $\|w_k\|_\infty < 1$. From (3.1) it follows that

$$w_k(i) = \frac{|a_i^T x_k - b(i)|}{\nu_k}, \quad i = 1, \dots, m,$$

for all $k \geq B$. Taking limits, it follows that $A\bar{x} = b$.

Suppose the penalty on infeasibility is

$$P(x) = \|Ax - b\|_\infty = \max_{\{w: \|w\|_1 \leq 1\}} w^T (Ax - b).$$

In this case, the smoothed penalty function

$$P_\nu(x) = \max_{\{w: \|w\|_1 \leq 1\}} \left\{ w^T (Ax - b) - \frac{\nu}{2} \|w\|_2^2 \right\},$$

has a Lipschitz continuous gradient

$$\nabla P_\nu(x) = A^T w_x$$

where

$$w_x = \Pi_{\ell_1}(1, (Ax - b)/\nu), \quad (3.2)$$

where Π_{ℓ_1} denotes the projection onto the ℓ_1 -ball.

Let \bar{x} denote any limit point of the sequence $\{x_k\}_{k \geq 1}$, i.e. there exists a subsequence $\{x_k\}_{k \in \mathcal{K}}$ such that $\lim_{k \rightarrow \infty} x_k = \bar{x}$. We will show that \bar{x} is feasible, i.e. $A\bar{x} = b$.

The stopping condition $\|\nabla Q_k(x_k)\| \leq \tau_k$ and the fact that A^T has full column rank implies that $\lim_{k \in \mathcal{K}} w_k = 0$, i.e. there exists B such that for all $k \geq B$, $\|w_k\|_1 < 1$. Since w_k is the optimal for the optimization problem defining $P_\nu(x)$, $\|w_k\|_1 < 1$ together with complementary slackness implies that

$$w_k = \frac{Ax_k - b}{\nu_k}.$$

Thus, $\lim_{k \rightarrow \infty} \|w_k\|_1 = 0$ implies that $A\bar{x} = b$.

3.2. Penalty methods for relaxed ℓ_1 -minimization problem. In this section we extend the results of the previous section to the relaxed ℓ_1 -minimization problem

$$\begin{aligned} & \text{minimize} && \|x\|_1, \\ & \text{subject to} && \|Ax - b\|_2 \leq \epsilon. \end{aligned} \tag{3.3}$$

To solve this problem we use the penalty function

$$\begin{aligned} P(x) &= \max \left\{ 0, \frac{1}{2} (\|Ax - b\|_2^2 - \epsilon^2) \right\}, \\ &= \max_{0 \leq t \leq 1} \left\{ \frac{t}{2} (\|Ax - b\|_2^2 - \epsilon^2) \right\}. \end{aligned}$$

Next, we smooth this function to get the smoothed penalty function

$$P_\nu(x) = \max_{0 \leq t \leq 1} \left\{ \frac{t}{2} (\|Ax - b\|_2^2 - \epsilon^2) - \frac{\nu}{2} t^2 \right\}. \tag{3.4}$$

This function is convex and its gradient is given by

$$\nabla P_\nu(x) = \phi_\nu \left(\frac{1}{2} \|Ax - b\|_2^2 - \frac{1}{2} \epsilon^2 \right) A^T (Ax - b),$$

where the function

$$\phi_\nu(y) = \begin{cases} 0 & y < 0, \\ \frac{y}{\nu}, & 0 \leq y \leq \nu, \\ 1 & y > \nu. \end{cases}$$

All the results in the previous section remain valid for this penalty function.

4. Numerical experiments.

4.1. Experimental setup. We tested SPA on randomly generated target signals. The target signal $x^* \in \mathbb{R}^n$ was chosen to be s -sparse, i.e. exactly s out of n components were nonzero. Without loss of generality we assume that n is an odd number. Following the experimental setup in a recent paper of Candés et al [15] we set

$$x^*(i) = \mathbf{1}(i \in \Lambda) \eta_1(i) 10^{5\eta_2(i)} \tag{4.1}$$

where

- (i) the set Λ was constructed by randomly selecting s indices from the set $\{1, \dots, n\}$,
- (ii) $\eta_1(i)$, $i \in \Lambda$, were independently, and identically distributed Bernoulli random variables taking values $+1$ or -1 with equal probability,
- (iii) $\eta_2(i)$, $i \in \Lambda$, were independently, and identically distributed uniform $[0, 1]$ random variables.

The signals x^* were created in this manner have a dynamic range of approximately 100dB.

The measurement matrix A and the measurement vector b were constructed as follows. We randomly selected $m = \frac{n}{8}$ frequencies from the set $\{0, \dots, \frac{(n-1)}{2}\}$. Let $C \in \mathbb{C}^{m \times n}$ denote a $m \times n$ partial Fourier matrix constructed from these randomly selected frequencies and $c = Cx^*$ denote the Fourier transform of the signal x^* evaluated at the chosen frequencies.

Let

$$\bar{A} = \begin{bmatrix} \Re(C) \\ \Im(C) \end{bmatrix}, \quad \bar{b} = \begin{bmatrix} \Re(c) \\ \Im(c) \end{bmatrix}$$

where $\Re(z) = \frac{1}{2}(z + z^*)$ (resp. $\Im(z) = \frac{1}{2}(z - z^*)$) denotes the real part (resp. imaginary part) of a complex number $z \in \mathbb{C}$, and the operators are applied component-wise. The matrix A and the vector b is constructed by removing linearly dependent rows from \bar{A} and the corresponding components from the vector \bar{b} .

We tested the algorithm for s -sparse signals with

- (i) three different sizes: small $n = 64 \times 64 - 1$, medium $n = 256 \times 256 - 1$, and large $n = 512 \times 512 - 1$,
- (ii) two sparsity levels: high $s = \lceil n/800 \rceil$, and low $s = \lceil n/40 \rceil$.

In order to assess the convergence properties of the SPA we used the stopping criterion

$$\|x_k - x^*\|_\infty \leq \epsilon. \quad (4.2)$$

We report results for $\epsilon = 1, 10^{-1}$ and 10^{-2} . The signal model in (4.1) and the stopping criterion implies that the algorithm produces x_k with $5 + \log_{10}(1/\epsilon)$ digits of accuracy. Note that the stopping criterion (4.2) is only used to test the convergence of the algorithm in this simulation study. In any real application x^* is not known and (4.2) cannot be used. We propose using the following stopping criterion.

$$|\|x_{k+1}\|_1 - \|x_k\|_1| \leq \epsilon \|x_k\|_1.$$

4.2. Implementation details. In this section, we describe the details of the stopping conditions and update schemes used in our numerical experiments.

τ_k **update.** The approximate optimality parameter τ_k was set as follows:

$$\begin{aligned} \tau_1 &\leftarrow c_0 \|\nabla Q_1(x_0)\|_2, \\ \tau_{k+1} &\leftarrow \min\{c_\tau \tau_k, c_0 \|\nabla Q_{k+1}(x_k)\|_2\}, \quad \text{for all } k \geq 1. \end{aligned}$$

We report results for $c_0 = 0.20$ and $c_\tau = 0.9$.

μ_k, ν_k **and** λ_k **updates.** Guided by the scaling result implicit in Theorem 2.4 we initialize $\lambda_1 = 1/\sqrt{n}$.

In each outer iteration k of the SPA, we solve a smoothed version of the penalized optimization problem

$$\begin{aligned} &\text{minimize} && \lambda_k \|x\|_1 + \|Ax - b\|_2, \\ &\text{subject to} && \|x\|_1 \leq \beta_k \end{aligned} \quad (4.3)$$

The dual of this optimization problem is given by

$$\begin{aligned} &\text{maximize} && -b^T w - \beta_k \|A^T w + \lambda_k u\|_\infty, \\ &\text{subject to} && \|u\|_\infty \leq 1, \\ &&& \|w\|_2 \leq 1. \end{aligned} \quad (4.4)$$

Nesterov's non-smooth optimization algorithm (see [14] for details) allows us to compute dual feasible iterates that converge to an approximately optimal dual solution. However, in order to reduce the total run time, the dual iterates can be computed once in every 10 inner iterations. Hence, the average number of costly matrix-vector multiplications is 2 per inner iteration.

Noting that q_0 and $(\hat{u}_0, \hat{w}_0) = (0, 0)$ are, respectively, primal and dual feasible, we initialize $\eta_0 = \lambda_1 \|q_0\|_1$. Let η_k denote the duality gap between the k -th primal iterate x_k and the dual iterates (\hat{u}_k, \hat{w}_k) returned by the Nesterov algorithm applied to the k -th subproblem.

We initialize $\mu_0 = \nu_0 = \infty$, and update parameters $(\mu_k, \nu_k, \lambda_k)$ for $k \geq 1$ as follows:

$$\begin{aligned} \delta_k &\leftarrow c_\eta \eta_{k-1}, \\ \mu_k &\leftarrow \min \left\{ c_\mu \mu_{k-1}, \frac{\delta_k}{(\sqrt{n} \lambda_k + 1) \sqrt{n}} \right\}, \\ \nu_k &\leftarrow \min \left\{ c_\nu \nu_{k-1}, \frac{\delta_k}{\sqrt{n} \lambda_k + 1} \right\}, \\ \lambda_{k+1} &\leftarrow c_\lambda \lambda_k. \end{aligned}$$

In the numerical results reported in this paper, the constants $c_\eta = 0.9$ and $c_\lambda = 0.95$.

Note that the parameter sequence $(\mu_k, \nu_k, \lambda_k, \delta_k)$ is *independent* of the problem dimension n . In our numerical experiments we tuned the parameter sequence on the $n = 64 \times 64 - 1$ problem and used these values for all the other problems.

Sparsity	ϵ	Table
$s = n/800$	1	Table 4.3
$s = n/800$	0.1	Table 4.5
$s = n/800$	0.01	Table 4.7
$s = n/40$	1	Table 4.2
$s = n/40$	0.1	Table 4.4
$s = n/40$	0.01	Table 4.6

TABLE 4.1
Summary of numerical experiments

4.3. Results. The Table 4.1 summarizes the sparsity conditions and the parameter settings that were investigated in the numerical experiments. The column marked **Table** lists the table where we display the results corresponding to the parameter setting of the particular row. For example, the results for sparsity pattern $s = n/40$, the error $\epsilon = 0.1$ are displayed in Table 4.4. In Tables 4.2–4.7, the row labeled **Iter** lists the total number of Nesterov inner iterations during the course of SPA. The rest of the row labels are self-explanatory. We generated $N = 10$ random instances for each of the experimental conditions. The column labeled **average** lists the average taken over the $N = 10$ random instances.

The experiment results support the following conclusions:

- (a) For a given sparsity type (high or low) and stopping criterion ϵ , the total number of Nesterov inner iterations is a very slowly growing function of the dimension n of the target signal.
- (b) Increasing the number of non-zero elements from $s = n/800$ to $s = n/40$ nearly doubled the total number of Nesterov iterations.

As remarked in Section 4.2 we used a fixed parameter sequence for all the experiments. We found that for a fixed measurement ratio m/n , sparsity ratio s/n , and solution accuracy ϵ , the total number of Nesterov iterations is effectively independent of the dimension n of the target signal. In our experiment we exploit this empirical result by first tuning the algorithm parameters for a smallest sized problem and subsequently using these fixed parameters for solving all larger problems.

TABLE 4.2
Experiment Results for $m = n/8$, $s = m/5$, $c_0 = 0.2$ and $\|x^{sol} - x^*\|_\infty \leq 1$

	n=512×512-1			n=256×256-1			n=64×64-1		
	Average	min	max	Average	min	max	Average	min	max
Iter. #	180.8	176	189	177.3	173	180	174.7	158	194
$\ x^{sol} - x^*\ _\infty / \ x^*\ _\infty$	5.6635E-06	1.8897E-06	1.1645E-05	5.2951E-06	4.5665E-06	5.8021E-06	8.0776E-06	4.9550E-06	2.2568E-05
$\ x^{sol} - x^*\ _\infty : x^*(i) > 0$	0.9836	0.9553	0.9966	0.9585	0.9270	0.9850	0.9717	0.9333	0.9939
$\ x^{sol} - x^*\ _\infty : x^*(i) = 0$	0.1104	0.0454	0.1848	0.0823	0.0444	0.1921	0.1804	0.0516	0.3532
$\ Ax^{sol} - b\ _2$	6.6386	3.4276	7.9532	3.8164	3.6326	4.0054	1.0137	0.2295	1.2920
$\ x^*\ _1$	56955136.2	53492728.4	58979210.8	14128997.7	12929129.5	15213917.9	947465.0	710794.2	1255575.2
Time	69.3	62.5	73.3	15.6	15.1	17.8	0.7	0.5	1.0

TABLE 4.3
Experiment Results for $m = n/8$, $s = m/100$, $c_0 = 0.2$ and $\|x^{sol} - x^*\|_\infty \leq 1$

	n=512×512-1			n=256×256-1			n=64×64-1		
	Average	min	max	Average	min	max	Average	min	max
Iter. #	102.2	87	111	102.5	86	110	82.4	61	97
$\ x^{sol} - x^*\ _\infty / \ x^*\ _\infty$	1.8042E-04	1.1350E-04	2.8708E-04	1.9752E-04	1.1655E-04	3.8065E-04	9.8089E-04	2.3135E-04	4.0182E-03
$\ x^{sol} - x^*\ _\infty : x^*(i) > 0$	0.9214	0.8138	0.9886	0.8902	0.7249	0.9906	0.8388	0.6713	0.9802
$\ x^{sol} - x^*\ _\infty : x^*(i) = 0$	0.0748	0.0329	0.1366	0.0779	0.0108	0.1459	0.0901	0.0174	0.2472
$\ Ax^{sol} - b\ _2$	1.6655	0.9652	2.3854	0.7985	0.4446	0.9964	0.3844	0.3021	0.5360
$\ x^*\ _1$	2857443.2	2164714.1	3525890.4	681155.5	248995.6	1035362.5	26374.3	2589.6	62557.6
Time	39.6	32.1	44.8	8.7	7.8	9.2	0.3	0.2	0.8

TABLE 4.4
Experiment Results for $m = n/8$, $s = m/5$, $c_0 = 0.2$ and $\|x^{sol} - x^*\|_\infty \leq 0.1$

n=512×512-1				n=256×256-1				n=64×64-1			
Average	min	max	Average	min	max	Average	min	max	Average	min	max
Iter. #	223.2	221	226	223.0	218	227	218.8	204	233		
$\ x^{sol} - x^*\ _\infty / \ x^*\ _\infty$	3.7992E-07	3.4301E-07	4.1582E-07	3.8903E-07	3.1312E-07	4.5356E-07	3.8759E-07	2.5096E-07	4.9158E-07		
$\ x^{sol} - x^*\ _\infty : x^*(i) > 0$	0.0935	0.0848	0.0994	0.0948	0.0880	0.0995	0.0954	0.0904	0.0991		
$\ x^{sol} - x^*\ _\infty : x^*(i) = 0$	0.0191	0.0136	0.0270	0.0210	0.0081	0.0390	0.0162	0.0051	0.0228		
$\ Ax^{sol} - b\ _2$	0.7517	0.7191	0.7837	0.3893	0.3597	0.4229	0.1148	0.0961	0.1309		
$\ x^*\ _1$	56955136.2	53492728.4	58979210.8	14128997.7	12929129.5	15213917.9	947465.0	710794.2	1255575.2		
Time	80.0	78.0	86.7	19.1	18.6	19.5	0.8	0.7	1.2		

TABLE 4.5
Experiment Results for $m = n/8$, $s = m/100$, $c_0 = 0.2$ and $\|x^{sol} - x^*\|_\infty \leq 0.1$

n=512×512-1				n=256×256-1				n=64×64-1			
Average	min	max	Average	min	max	Average	min	max	Average	min	max
Iter. #	135.0	112	153	131.7	100	159	118.3	86	233		
$\ x^{sol} - x^*\ _\infty / \ x^*\ _\infty$	2.4361E-05	1.0785E-05	3.5758E-05	2.7913E-05	1.1940E-05	5.0021E-05	9.5154E-05	3.0493E-05	4.6618E-04		
$\ x^{sol} - x^*\ _\infty : x^*(i) > 0$	0.0954	0.0882	0.0998	0.0907	0.0675	0.0983	0.0944	0.0801	0.0995		
$\ x^{sol} - x^*\ _\infty : x^*(i) = 0$	0.0103	0.0011	0.0297	0.0061	0.0028	0.0144	0.0037	0.0011	0.0118		
$\ Ax^{sol} - b\ _2$	0.2058	0.1059	0.2986	0.1071	0.0617	0.1653	0.0352	0.0208	0.0454		
$\ x^*\ _1$	2857443.2	2164714.1	3525890.4	681155.5	248995.6	1035362.5	26374.3	2589.6	62557.6		
Time	49.5	43.3	55.9	11.1	8.5	13.5	0.4	0.3	0.8		

TABLE 4.6
Experiment Results for $m = n/8$, $s = m/5$, $c_0 = 0.2$ and $\|x^{sol} - x^*\|_\infty \leq 0.01$

	n=512×512-1			n=256×256-1			n=64×64-1		
	Average	min	max	Average	min	max	Average	min	max
Iter. #	271.3	268	277	270.8	266	275	269.8	252	291
$\ x^{sol} - x^*\ _\infty / \ x^*\ _\infty$	2.3925E-08	1.8388E-08	2.6919E-08	2.2858E-08	1.8023E-08	2.8018E-08	1.8854E-08	5.3290E-09	3.2209E-08
$\ x^{sol} - x^*\ _\infty : x^*(i) > 0$	0.0095	0.0090	0.0100	0.0095	0.0088	0.0099	0.0095	0.0092	0.0099
$\ x^{sol} - x^*\ _\infty : x^*(i) = 0$	0.0019	0.0010	0.0028	0.0023	0.0015	0.0028	0.0018	0.0005	0.0033
$\ Ax^{sol} - b\ _2$	0.0722	0.0697	0.0750	0.0375	0.0350	0.0400	0.0104	0.0076	0.0124
$\ x^*\ _1$	56955136.2	53492728.4	58979210.8	14128997.7	12929129.5	15213917.9	947465.0	710794.2	1255575.2
Time	96.0	94.4	98.2	24.6	22.7	27.0	1.0	0.8	1.4

TABLE 4.7
Experiment Results for $m = n/8$, $s = m/100$, $c_0 = 0.2$ and $\|x^{sol} - x^*\|_\infty \leq 0.01$

	n=512×512-1			n=256×256-1			n=64×64-1		
	Average	min	max	Average	min	max	Average	min	max
Iter. #	154.4	119	176	160.4	140	176	172.3	106	253
$\ x^{sol} - x^*\ _\infty / \ x^*\ _\infty$	2.3698E-06	1.6287E-06	4.5832E-06	3.0396E-06	1.4318E-06	5.2729E-06	1.0964E-05	2.6672E-06	4.5507E-05
$\ x^{sol} - x^*\ _\infty : x^*(i) > 0$	0.0092	0.0077	0.0100	0.0094	0.0075	0.0100	0.0091	0.0079	0.0098
$\ x^{sol} - x^*\ _\infty : x^*(i) = 0$	0.0010	0.0005	0.0024	0.0009	0.0001	0.0024	0.0007	0.0002	0.0015
$\ Ax^{sol} - b\ _2$	0.0198	0.0156	0.0252	0.0108	0.0075	0.0154	0.0041	0.0035	0.0045
$\ x^*\ _1$	2857443.2	2164714.1	3525890.4	681155.5	248995.6	1035362.5	26374.3	2589.6	62557.6
Time	55.6	43.0	62.4	14.6	11.7	16.7	0.6	0.4	0.8

5. Conclusion. In this paper, we proposed a smoothed penalty algorithm (SPA) for the sparse recovery problem. The SPA recovers the target signal by solving a sequence of smoothed penalized sub-problems, and each sub-problem is solved using Nesterov's optimal method for simple sets [13, 14]. We show that the continuation scheme used in SPA provably converges to the target signal. Since we penalize infeasibility by the exact penalty function $\|Ax - b\|$, where $\|\cdot\|$ can be ℓ_1 , ℓ_2 or ℓ_∞ norm, an accurate solution is obtained before penalty parameter takes on arbitrarily small value; consequently, our proposed algorithm is numerically stable. We found that for a fixed measurement ratio m/n , sparsity ratio s/n , and solution accuracy ϵ , the total number of Nesterov iterations is effectively independent of the dimension n of the target signal; thus, one can tune the parameters on the smallest problem and use these parameters for all larger problems. The numerical results reported in this paper show that SPA required very few iterations to accurately recover the target signal.

REFERENCES

- [1] N. S. AYBAT AND A. CHAKRABORTY, *Compressed computerized tomographic imaging*, tech. report, Siemens Corp. Research, 2009.
- [2] E. CANDÈS AND J. ROMBERG, *Quantitative robust uncertainty principles and optimally sparse decompositions*, Foundations of Computational Mathematics, 6 (2006), pp. 227–254.
- [3] E. CANDÈS, J. ROMBERG, AND T. TAO, *Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information*, IEEE Trans. Info. Th., 52 (2006).
- [4] E. CANDÈS AND T. TAO, *Near optimal signal recovery from random projections: universal encoding strategies?*, IEEE Trans. Info. Th., 52 (2006), pp. 5406–5425.
- [5] E. VAN DEN BERG AND M. P. FRIEDLANDER, *Probing the pareto frontier for basis pursuit solutions*, SIAM Journal on Scientific Computing, 31 (2008), pp. 890–912.
- [6] D. DONOHO, *Compressed sensing*, IEEE Trans. Info. Th., 52 (2006), pp. 1289–1306.
- [7] W. YIN E. T. HALE AND Y. ZHANG, *A fixed-point continuation for ℓ_1 -regularized minimization with applications to compressed sensing*, tech. report, Rice University, 2007.
- [8] W. YIN E. T. HALE AND Y. ZHANG, *Fixed-point continuation for ℓ_1 -minimization: Methodology and convergence*, SIAM Journal on Optimization, 19 (2008), pp. 1107–1130.
- [9] M. FORNASIER I. DAUBECHIES AND I. LORIS, *Accelerated projected gradient method for linear inverse problems with sparsity constraints*, Journal of Fourier Analysis and Applications, 14 (2008), pp. 764–792.
- [10] Y. SINGER J. DUCHI, S. SHALEW-SHWARTZ AND T. CHANDRA, *Efficient projections onto the ℓ_1 -ball for learning in high dimensions*, in Proceedings, Twenty-Fifth International Conference on Machine Learning, Andrew McCallum and Sam Roweis, eds., Helsinki, Finland, 2008, pp. 272–279.
- [11] S. J. KIM K. KOH AND S. BOYD, *Solver for ℓ_1 -regularized least squares problems*, tech. report, Stanford University, 2007.
- [12] R. NOWAK M. A. FIGUEIREDO AND S. J. WRIGHT, *Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems*, IEEE Journal of Selected Topics in Signal Processing, 1 (2007), pp. 586–597.
- [13] YU. NESTEROV, *Introductory Lectures on Convex Optimization: A Basic Course*, Kluwer Academic Publishers, 2004.
- [14] YU. NESTEROV, *Smooth minimization of nonsmooth functions*, Mathematical Programming, 103 (2005), pp. 127–152.
- [15] J. BOBIN S. BECKER AND E. CANDÈS, *Nesta: a fast and accurate first-order method for sparse recovery*. Submitted for publication, April 2009.
- [16] D. GOLDFARB W. YIN, S. OSHER AND J. DARBON, *Bregman iterative algorithms for ℓ_1 minimization with applications to compressed sensing*, SIAM Journal on Imaging Sciences, 1 (2008), pp. 143–168.
- [17] D. GOLDFARB Z. WEN, W. YIN AND Y. ZHANG, *A fast algorithm for sparse reconstruction based on shrinkage, subspace optimization and continuation*, tech. report, Columbia University, 2009.

Appendix A. Details of the steps in SPA. In this section, we collect together results that show that SPA is very efficient as long as one can compute the matrix-vector products Ax and $A^T y$ efficiently.

A.1. $\|A\|_2$ when measurements are discrete Fourier transforms. Let $C \in \mathbb{C}^{m \times n}$ be a partial Fourier matrix, where rows of C are chosen randomly among the rows of n dimensional Fourier matrix. Without loss of generality, assume that n is an odd number. Since the target signal x^* takes real values, we can restrict the set of m randomly selected frequencies $\Gamma \subset \{0, 1, \dots, \frac{n-1}{2}\}$ without any loss of generality.

Let $A_R = \Re(C)$, $A_I = \Im(C)$ and define

$$\bar{A} = \begin{bmatrix} A_R \\ A_I \end{bmatrix}.$$

Let $a^R(k)$ and $a^I(k)$ that denote the rows in A_R and in A_I , respectively, corresponding to the frequency index $k \in \Gamma$. Then

$$\begin{aligned} a^R(k) &= \left[\frac{1}{\sqrt{n}} \cos\left(\frac{2\pi jk}{n}\right) \right]_{j=0, \dots, n-1} \\ a^I(k) &= - \left[\frac{1}{\sqrt{n}} \sin\left(\frac{2\pi jk}{n}\right) \right]_{j=0, \dots, n-1}. \end{aligned}$$

Using simple properties of trigonometric sequences it is easy to establish that for all $k, l \in \Gamma$,

$$\begin{aligned} a^R(k)a^I(l)^T &= 0 \\ a^R(k)a^R(l)^T &= \begin{cases} 0, & \text{if } l \neq k, \\ \frac{1}{2}, & \text{if } l = k > 0; \\ 1, & \text{if } l = k = 0; \end{cases} \\ a^I(k)a^I(l)^T &= \begin{cases} 0, & \text{if } l \neq k, \\ \frac{1}{2}, & \text{if } l = k > 0; \\ 0, & \text{if } l = k = 0; \end{cases} \end{aligned} \tag{A.1}$$

The measurement matrix A is obtained by removing the row $a^I(0)$ from \bar{A} if $0 \in \Gamma$; otherwise A is set to \bar{A} . Since the vector $a^I(0) = 0$, removing $a^I(0)$ does result in any loss of generality. Furthermore, (A.1) implies that AA^T is a diagonal matrix with entries taking values in the set $\{\frac{1}{2}, 1\}$. Thus, $\sigma_{\min}(A) = \frac{1}{\sqrt{2}}$ and $\|A\|_2 = \sigma_{\max}(A) = 1$.

A.2. ℓ_2 or Least squares projection. In this section, we show that the ℓ_2 -projection

$$\begin{aligned} &\text{minimize} && \|x - \hat{x}\|_2^2, \\ &\text{subject to} && Ax = b. \end{aligned} \tag{A.2}$$

can be computed efficiently. We compute this projection several times during the course of SPA. We initialize SPA by setting $x_0 = \operatorname{argmin}\{\|x\|_2 \mid Ax = b\}$. Along the course of the algorithm we update β by solving $\min\{\|x - x_k\|_2 \mid Ax = b\}$. We show that both these problems can be solved efficiently. Note that in each instance of projection problem encountered during SPA, only the vector \hat{x} changes but the matrix A remains constant.

We first consider the special case where A corresponds to partial Fourier matrix measurements. Without loss of generality, assume that n is an odd number. In the Fourier case, the measurement matrix A is constructed as follows. Since the target signal x^* only takes real values, the set of m randomly selected frequencies $\Gamma \subset \{0, 1, \dots, \frac{n-1}{2}\}$ without any loss of generality. Let $A_R = \Re(C)$ and $b_R = \Re(b)$ denote the real part of the matrix C and the vector b , respectively, and $A_I = \Im(C)$, $\bar{b}_I = \Im(b)$ denote the imaginary part of the matrix C and the vector b , respectively. Let

$$\bar{A} = \begin{bmatrix} A_R \\ A_I \end{bmatrix}, \quad \bar{b} = \begin{bmatrix} b_R \\ b_I \end{bmatrix} \tag{A.3}$$

The measurement matrix A and the righthand-side vector b are constructed by removing redundant rows from the matrix \bar{A} and the corresponding components from the vector \bar{b} .

Since redundant equations do not alter the feasible region, i.e. $X = \{x \in \mathbb{R}^n | Ax = b\} = \{x \in \mathbb{R}^n | \bar{A}x = \bar{b}\}$, it follows that

$$\begin{aligned} & \text{minimize} && \frac{1}{2} \|x - \hat{x}\|_2^2, \\ & \text{subject to} && Ax = b, \end{aligned}$$

is equivalent to

$$\begin{aligned} & \text{minimize} && \frac{1}{2} \|x - \hat{x}\|_2^2, \\ & \text{subject to} && A_R x = b_R, \\ & && A_I x = b_I. \end{aligned} \tag{A.4}$$

The optimal solution x^* to (A.4) satisfies the KKT conditions

$$\begin{aligned} A_R^T \lambda_R + A_I^T \lambda_I - x^* &= -\hat{x}, \\ A_R A_R^T \lambda_R + A_R A_I^T \lambda_I &= b_R - A_R \hat{x}, \\ A_I A_R^T \lambda_R + A_I A_I^T \lambda_I &= b_I - A_I \hat{x}, \end{aligned} \tag{A.5}$$

for some $\lambda_R, \lambda_I \in \mathbb{R}^m$.

From Section A.1, $a^R(k)a^I(l)^T = 0$ for all k and l , i.e. $A_I A_R^T = A_R A_I^T = 0$. Thus, the KKT conditions (A.5) simply to

$$\begin{aligned} A_R^T \lambda_R + A_I^T \lambda_I - x^* &= -\hat{x}, \\ A_R A_R^T \lambda_R &= b_R - A_R \hat{x}, \\ A_I A_I^T \lambda_I &= b_I - A_I \hat{x}. \end{aligned} \tag{A.6}$$

The vectors $A_R \hat{x}$ and $A_I \hat{x}$ can be computed via a single FFT of \hat{x} requiring $\mathcal{O}(n \log(n))$ operations. Since the matrix products $A_R A_R^T$ and $A_I A_I^T$ are diagonal (see Section A.1), we can compute λ_R and λ_I in $\mathcal{O}(m)$ operations. Next, $x^* = \hat{x} + A_R \lambda_R + A_I \lambda_I$ can be computed by one inverse FFT using $\mathcal{O}(n \log(n))$ operations. Thus, computing x requires $2\mathcal{O}(n \log(n)) + \mathcal{O}(m)$ operations.

Next, we consider the special case when A is a real matrix with orthonormal rows. This is the case when the measurement vector b corresponds to a sampled Discrete Cosine Transforms. Then

$$x^* = \operatorname{argmin}\{\|x - \hat{x}\|_2 \mid Ax = b\} = \hat{x} + A^T(b - A\hat{x}).$$

Hence, if multiplications with A and A^T can be computed efficiently, the optimal projection x^* can be computed efficiently. When A is a partial DCT matrix, x^* can be computed by solving one forward and one inverse DCT, i.e. in $2\mathcal{O}(n \log(n))$ operations.

Next, consider the case when A is a real matrix with full row rank such that matrix-vector multiplications with A and A^T can be computed efficiently. Suppose Ax can be computed in $\kappa_f(m, n)$ operations and $A^T y$ can be computed in $\kappa_r(m, n)$ operations. In this case the optimal solution x^* satisfy the KKT conditions

$$\begin{aligned} x^* &= \hat{x} + A^T \lambda, \\ AA^T \lambda &= b - A\hat{x}, \end{aligned}$$

for some $\lambda \in \mathbb{R}^m$. Since A is assumed to have full row rank, $(AA^T)^{-1} \in \mathbb{R}^{m \times m}$ exists. Hence, $x^* = \hat{x} + A^T(AA^T)^{-1}(b - A\hat{x})$. We compute x^* efficiently as follows. Let $A = U\Sigma V^T$ denote the singular value decomposition (SVD) of A where $\Sigma = \operatorname{diag}(\sigma_1, \sigma_2, \dots, \sigma_m)$, with $\sigma_i > 0$ for all $i = 1, \dots, m$, and $U, V \in \mathbb{R}^{n \times m}$ such that $U^T U = I$, $V^T V = I$. Computing the SVD takes $\mathcal{O}(m^2 n)$ operations.

The KKT multipliers $\lambda = \sum_{i=1}^m \left(\frac{u_i^T (b - A\hat{x})}{\sigma_i^2} \right) u_i$. Thus,

$$x^* = \hat{x} + \sum_{i=1}^m \left(\frac{u_i^T (b - A\hat{x})}{\sigma_i^2} \right) A^T u_i.$$

ℓ_1 -PROJECTION (\hat{x}, σ)

$x_s \leftarrow \hat{x}$ sorted in increasing order

Compute $\rho = \max \left\{ j \in \{1, \dots, n\} : x_s(j) - \frac{1}{j} \left(\sum_{r=1}^j x_s(r) - \sigma \right) > 0 \right\}$

$\xi \leftarrow \frac{1}{\rho} \left(\sum_{i=1}^{\rho} x_s(i) - \sigma \right)$

$x^*(i) \leftarrow \max\{\hat{x}(i) - \xi, 0\}, i = 1, \dots, n$

return x^*

FIG. A.1. *Projection onto the simplex*

We compute and store the values $\{A^T u_i\}_{i=1, \dots, m}$ at the beginning of the algorithm, which requires $\mathcal{O}(m\kappa_r(m, n))$ operations. Given the precomputed values of $\{A^T u_i\}_{i=1, \dots, m}$, x^* can be computed in $\mathcal{O}(m(m+n) + \kappa_f(m, n))$ operations.

A.3. Projection onto the ℓ_1 -ball. In this section we show that the Euclidean projection problem

$$\begin{aligned} & \text{minimize} && \|x - \hat{x}\|_2^2, \\ & \text{subject to} && \|x\|_1 \leq \sigma, \end{aligned} \tag{A.7}$$

can be computed efficiently.

Define $\hat{y} = |\hat{x}|$ and consider the following optimization problem.

$$\begin{aligned} & \text{minimize} && \|y - \hat{y}\|_2^2, \\ & \text{subject to} && \sum_{i=1}^n y_i \leq \sigma, \\ & && y \geq 0. \end{aligned} \tag{A.8}$$

Let x^* denote the projection of \hat{x} onto the ℓ_1 -ball, i.e. the optimal solution of (A.7). Then it is easy to check that $x^*(i)\hat{x}(i) \geq 0$ for all $i = 1, \dots, n$, i.e. $x^*(i) = \text{sign}(\hat{x}(i)) |x^*(i)|$, for all $i = 1, \dots, n$. This implies that

$$x^*(i) = \text{sign}(\hat{x}(i))y^*(i), \quad i = 1, \dots, n,$$

where y^* is the projection of \hat{y} onto the simplex, i.e. the optimal solution of (A.8). Thus, the optimal solution of (A.7) can be recovered from the optimal solution of (A.8). Singer et al [10] show that the algorithm in Figure A.1 computes an optimal solution to (A.8) in $\mathcal{O}(n \log(n))$ operations.