• ARTICLES • doi:

Adaptive Projected Gradient Thresholding Methods for Constrained l_0 Problems

ZHAO ZhiHua[1], XU FengMin[1,*] & LI XiangYang[1]

[1] *Department of Mathematics and Statistics, Xi'an Jiaotong University, Xi'an* 710049, *China*
Email:zhaozhihua@stu.xjtu.edu.cn, fengminxu@mail.xjtu.edu.cn, llxxyy0522@163.com

Received ; accepted

Abstract In this paper, we propose and analyze Adaptive Projected Gradient Thresholding (APGT) methods for finding sparse solutions of the underdetermined linear system with equality and box constrains. The general convergence will be demonstrated, and in addition the bound of the number of iterations is established in some special cases. Under suitable assumptions, it is proved that any accumulation point of the sequence generated by the APGT methods is a local minimizer of the underdetermined linear system. Moreover, the APGT methods, under certain conditions, can find all s-sparse solutions for accurate measurement cases and guarantee the stability and robustness for flawed measurement cases. Numerical examples are presented to show the accordance with theoretical results in compressed sensing and verify high out-of-sample performance in index tracking.

Keywords projected gradient, l_0 constraints, compressed sensing, index tracking, hard thresholding

MSC(2010) 90C52, 90C26

1 Introduction

Sparse optimization has attracted a great deal of attention over recent years in many fields, which aims at finding sparse solutions of a system or an equation. For example, Compressed Sensing [1], Index Tracking [2] in finance optimization and many other important applications, such as image reconstruction, image restoration, supervised learning, unsupervised learning and statistical inference. In this paper, we consider the sparse solutions of the underdetermined linear system with equality and box constraints by solving the following problem

$$\begin{aligned} &\min \|y - Ax\|_2^2 \\ &s.t. \;\; e^T x = d; \\ &\qquad 0 \leqslant x \leqslant u; \\ &\qquad \|x\|_0 \leqslant s, \end{aligned} \tag{1.1}$$

where matrix $A \in R^{m\times n}, y \in R^m, x \in R^n (m \ll n)$. Generally, this problem is considered to be NP-hard and hence is difficult to solve. One special case of (1.1) is the unconstrained sparse least squares problem.

There have been many kinds of algorithms proposed for sparse optimization problems. For the unconstrained l_0 problems, the existing algorithms can be classified into three categories. The first kind of methods is greedy algorithms, mainly including a variety of matching pursuit (MP) algorithms. Tropp and Gilbert [3] proposed orthogonal matching pursuit (OMP) algorithms with orthogonal technique to avoid number of iterations on the basis of MP algorithms. However, both MP and OMP algorithms have

*Corresponding author

©

large cost of computations and slow rate of convergence. Donoho et al. [4] gave the stagewise matching pursuit (StOMP) method to improve the convergence speed of MP algorithms. Needell and Tropp [5] proposed the compressive sampling matching pursuit (CoSaMP) algorithm, which was robust to noise. The second kind of methods is iterative projection methods, mainly including iterative thresholding algorithm. Blumensath and Davies [6] established the iterative hard thresholding (IHT) algorithm to solve the approximation of l_0 problem. Under some conditions, IHT method can solve the l_0 problem efficiently. Recently, Foucart [7] proposed hard thresholding pursuit (HTP) algorithm by combining the CoSaMP and IHT algorithms, and gave the general convergence analysis. The third kind of methods is the relaxation methods, including convex relaxation l_1 problems and non-convex relaxation $l_p(0 < p < 1)$ problems. As for l_1 problems, there exist many exclusive and efficient algorithms in [8–15]. As for l_p problems, there are also many works [16–18] and especially an iterative thresholding algorithm [19] for $l_p(p = 1/2)$ problem. For constrained l_0 problems, Lu and Zhang [21] proposed the penalty decomposition method. In 2013, Kyrillidis et al. [22] gave the projected gradient method onto cardinality constraints. Constrained l_0 problems can also be tackled by solving the convex relaxation l_1 problems. Kim et al. [20] transformed the LASSO [23] into quadratic programming solved by the interior point method. Moreover, Candes et al. [24] changed the LASSO into quadratic conic programming which can be solved by interior point methods or Newton methods.

Motivated by HTP method, we propose adaptive projected gradient thresholding (APGT) methods, which its general convergence and a finite number of iterations in particular are also demonstrated. At each iteration, these methods usually solve several projected gradient subproblems which possess closed-form solutions and some least squares subproblems. To deal with (1.1), we first consider the box constrained l_0 sparse minimization problem and apply the APGT methods to solve it. Under some suitable assumptions, we establish that any accumulation point of the sequence generated by the APGT methods is a local minimizer of the original problem. Furthermore, the APGT methods indeed find all s-sparse solutions for accurate measurement cases and guarantee stability and robustness for flawed measurement cases. Then, we find that the above conclusions are still valid for (1.1). Finally, we conduct empirical tests to show methods are in accordance with theoretical results in compressed sensing and verify the good performance in real index tracking problems.

The rest of the paper is organized as follows. In Section 1.1, we give some notations which will be used in this paper. In Section 2, we propose adaptive projected gradient thresholding methods for the l_0 sparse minimization problem with box constraints and establish the convergence. In Section 3, we extend the APGT methods for solving problem (1.1). In Section 4, we present some empirical tests in signal recovery and index tracking with comparison to other existed approaches respectively. Finally, some concluding remarks are made in Section 5.

1.1 Notations

Given a matrix $A \in R^{m\times n}$, denote σ_{max} and σ_{min} the largest and smallest singular values of matrix A respectively. Define $cond(A) = \sigma_{max}/\sigma_{min}$ and $\|A\|_2 = \sup_{x\neq 0} \|Ax\|_2/\|x\|_2$. Let $I_s(x)$ be the indices of s largest entries of vector x and $\Pi_{[0,u]}(t)$ be the projection of $t \in R$ on interval $[0, u]$, that is,

$$\Pi_{[0,u]}(t) = \begin{cases} 0, & \text{if } t \leqslant 0; \\ t, & \text{if } 0 < t < u; \\ u, & \text{if } t \geqslant u. \end{cases} \qquad \forall t \in \Re.$$

Define Σ_s the set of all s indices set, that is, $\Sigma_s = \{I : |I| \leqslant s, I \subset \{1, 2, \cdots , n\}\}$. For index sets $S_1, S_2 \subset \{1, 2, \cdots , n\}$, $\overline{S}_1$ is the complementary set of S_1 and $S_1\backslash S_2$ is the difference set between S_1 and S_2, that is, $S_1\backslash S_2 = \{x \in R | x \in S_1, x \notin S_2\}$. Define x_S the vector obtained by keeping the entries of x indexed in S and setting the other components to zeroes.

2 l_0 sparse problem with box constraints

In this section, we consider the following l_0 sparse minimization problem with box constraints:

$$\begin{aligned} &\min \|y - Ax\|_2^2 \\ &s.t. \ \ x \in \Omega_1, \end{aligned} \tag{2.1}$$

where $\Omega_1 = \{x \in R^n : 0 \leqslant x \leqslant u, \|x\|_0 \leqslant s\}$.

2.1 Algorithm and convergence analysis

In this subsection, we propose adaptive projected gradient thresholding (APGT) methods for solving problem (2.1), including adaptive projected gradient algorithm with line search (APGT-LS) and algorithm with constant step (APGT-C). We shall prove the accumulation point of the sequence generated by these methods is a local minimizer of (2.1).

Before proceeding, we denote the surrogate function at iteration point x^k:

$$C_\mu(x, x^k) = \mu(\|y - Ax\|_2^2 - \|Ax - Ax^k\|_2^2) + \|x - x^k\|_2^2.$$

Clearly, it is an upper-bound approximation of the original objective function at the point x^k, that is,

$$\begin{aligned} C_\mu(x, x^k) \ &\geqslant \mu\|y - Ax\|_2^2 - \mu\|A\|_2^2\|x - x^k\|_2^2 + \|x - x^k\|_2^2 \\ &= \mu\|y - Ax\|_2^2 + (1 - \mu\|A\|_2^2)\|x - x^k\|_2^2 \\ &\geqslant \mu\|y - Ax\|_2^2, \end{aligned}$$

provided $\mu\|A\|_2^2 \leqslant 1$. Define $S_\mu(x^k) = x^k + \mu A^*(y - Ax^k)$, then

$$\arg\min_{x \in R^n} C_\mu(x, x^k) = \arg\min_{x \in R^n} \|x - S_\mu(x^k)\|_2^2.$$

The main idea of the APGT methods consists of two parts: the first one is to determine the iterative indices sets determined by minimizing an surrogate function; the second one is to solve the least squares subproblem based on the determined index sets. The APGT methods are described as follows:

Adaptive projected gradient thresholding algorithm with line search (APGT-LS).
Let $0 < \mu_{min} < \mu_{max} \leqslant 1/\sigma_{max}^2, c > 0$. Choose $\mu_0 \in [\mu_{min}, \mu_{max}]$, $\tau \in (0, 1)$, and an arbitrary $x^0 \in \Omega_1$ and set $k = 0$.
1) Solve

$$\begin{aligned} S^{k+1} &= I_s(S_{\mu_k}(x^k)), \\ p^{k+1} &\in \arg\min_{x \in \Omega_1} \|x - S_{\mu_k}(x^k)\|_2^2, \end{aligned} \tag{2.2}$$

$$x^{k+1} \in \arg\min\{\|y - Ax\|_2^2 : 0 \leqslant x \leqslant u, Supp(x) \subset S^{k+1}\}; \tag{2.3}$$

2) If

$$\|y - Ax^{k+1}\|_2^2 - \|y - Ax^k\|_2^2 \leqslant -c\|p^{k+1} - x^k\|_2^2, \tag{2.4}$$

set $k \leftarrow k + 1$ and go to step 1);
Otherwise, set $\mu_k = \tau\mu_k$ and go to step 1).

Remarks
(i) Let $\mu_k \equiv 1$, the above algorithm has constant steps, which is called as **APGT-C algorithm**;
(ii) A specific choice of μ_k is given by the following formula proposed by Blumensath and Davies [25]:

$$\mu_k = \Pi_{[\mu_{min}, \mu_{max}]}\left(\frac{\|(A^*(y - Ax^k))_{S^k}\|_2^2}{\|A(A^*(y - Ax^k))_{S^k}\|_2^2}\right);$$

(iii) If the subproblem (2.3) is too costly, we may consider an inexact solution and take a certain number k of gradient descent iterations ;

(iv) According to the property of the above algorithm, we can set the termination criterion of APGT methods to be $S^{k+1} = S^k$; another practical termination criterion is to set either a maximum number of iterations or the absolute (or relative) error in the objective function value.

Next, we will give the closed-form solution of the subproblem (2.2) and prove the line search condition (2.4) can be satisfied within a limited number of steps.

Firstly, the subproblem (2.2) is a special case of the more general problem

$$\min_{x\in\Omega_1} \|x-a\|_2^2,$$

which has a closed-form solution for any $a \in R^n$. See Lemma 2.1.

Lemma 2.1. *The optimal solution of problem* $\min\limits_{x\in\Omega_1} \|x-a\|_2^2$, $x^*(a)$, *has the closed-form*

$$[x^*(a)]_i = \begin{cases} [\Pi_{[0,u]}(a)]_i, & i \in I_s(a); \\ 0, & otherwise. \end{cases}$$

Proof. For a fixed index set I satisfying $I \in \Sigma_s$, the original problem can be expressed as

$$\begin{aligned} &\min \|(x-a)_I\|_2^2 + \|(a)_{I^C}\|_2^2 \\ &s.t.\ \ 0 \leqslant x_i \leqslant u, \quad i \in I, \\ &\qquad x_i = 0, \quad i \in \overline{I}, \end{aligned}$$

and the optimal solution is

$$x_i = \begin{cases} [\Pi_{[0,u]}(a)]_i, & i \in I; \\ 0, & otherwise. \end{cases}$$

Thus, the original problem is equivalent to find an optimal index set in Σ_s, that is,

$$\begin{aligned} & \arg\min_{I\in\Sigma_s} \|(x-a)_I\|_2^2 + \|(a)_{\overline{I}}\|_2^2 \\ = \ & \arg\min_{I\in\Sigma_s} \|(\Pi_{[0,u]}(a)-a)_I\|_2^2 + \|(a)_{\overline{I}}\|_2^2 \\ = \ & \arg\min_{I\in\Sigma_s} \|(\Pi_{[0,u]}(a)-a)_I\|_2^2 - \|(a)_I\|_2^2 + \|a\|_2^2 \\ = \ & \arg\max_{I\in\Sigma_s} \|(a)_I\|_2^2 - \|(\Pi_{[0,u]}(a)-a)_I\|_2^2 \\ = \ & \arg\max_{I\in\Sigma_s} \sum_{i\in I}(a_i^2 - (\Pi_{[0,u]}(a_i)-a_i)^2). \end{aligned}$$

Therefore, the problem $\min\limits_{x\in\Omega_1} \|x-a\|_2^2$ is equivalent to finding an optimal index set I^* such that

$$I^* \in \arg\max_{I\in\Sigma_s} \sum_{i\in I}(a_i^2 - (\Pi_{[0,u]}(a_i)-a_i)^2).$$

Define $\phi(t) = t^2 - (\Pi_{[0,u]}(t)-t)^2$. Then $\phi(t) = 0, \forall t < 0$. Furthermore, $\phi'(t) = 2\Pi_{[0,u]}(t) \geqslant 0$, that is, $\phi(t)$ is a non-decreasing function. Hence, $I^* = I_s(a)$ is an optimal index set and

$$[x^*(a)]_i = \begin{cases} [\Pi_{[0,u]}(a)]_i, & i \in I_s(a); \\ 0, & otherwise \end{cases}$$

is an optimal solution. $\square$

Secondly, the line search condition can be satisfied in finite steps for each iteration of APGT-LS algorithm.

Theorem 2.2. *For each* $k \geqslant 0$, *the line search condition (2.4) is satisfied after at most* $\lceil \frac{-\log(\mu_{max})-\log(\|A\|_2^2+c)}{\log(\tau)} + 2\rceil$ *steps.*

Proof. According to the definition of p^{k+1}, we easily know $C_{\mu_k}(p^{k+1},x^k)\leqslant C_{\mu_k}(x,x^k),\forall x\in\Omega_1$. In addition,

$$\begin{aligned}C_{\mu_k}(p^{k+1},x^k) &= \mu_k(\|y-Ap^{k+1}\|_2^2-\|Ap^{k+1}-Ax^k\|_2^2)+\|p^{k+1}-x^k\|_2^2\\ &\geqslant \mu_k\|y-Ap^{k+1}\|_2^2-\mu_k\|A\|_2^2\|p^{k+1}-x^k\|_2^2+\|p^{k+1}-x^k\|_2^2\\ &= \mu_k\|y-Ap^{k+1}\|_2^2+(1-\mu_k\|A\|_2^2)\|p^{k+1}-x^k\|_2^2.\end{aligned}$$

Notice that p^{k+1} is a feasible solution of problem $\{\min\|y-Ax\|_2^2:0\leqslant x\leqslant u,Supp(x)\subset S^{k+1}\}$. Then

$$\mu_k\|y-Ax^{k+1}\|_2^2\leqslant\mu_k\|y-Ap^{k+1}\|_2^2. \tag{2.5}$$

This yields

$$\begin{aligned}\mu_k\|y-Ax^{k+1}\|_2^2 &\leqslant \mu_k\|y-Ap^{k+1}\|_2^2+(1-\mu_k\|A\|_2^2)\|p^{k+1}-x^k\|_2^2\\ &\leqslant C_{\mu_k}(p^{k+1},x^k)\\ &\leqslant C_{\mu_k}(x^k,x^k)=\mu_k\|y-Ax^k\|_2^2.\end{aligned}$$

Hence,

$$\|y-Ap^{k+1}\|_2^2-\|y-Ax^k\|_2^2\leqslant(\|A\|_2^2-\frac{1}{\mu_k})\|p^{k+1}-x^k\|_2^2. \tag{2.6}$$

Combining (2.5) and (2.6), we obtain

$$\|y-Ax^{k+1}\|_2^2-\|y-Ax^k\|_2^2\leqslant(\|A\|_2^2-\frac{1}{\mu_k})\|p^{k+1}-x^k\|_2^2.$$

Define $c_{max}=1/\mu_{min}-\|A\|_2^2$ and $c_{min}=1/\mu_{max}-\|A\|_2^2$. It is not hard to notice that (2.4) holds if $c\in(c_{min},c_{max})$, which implies that a satisfied μ_k can been found in finite number of updating. Let $\overline{\mu}_k$ be the final value of μ_k at the kth outer iteration. Thus, we have $\tau/\overline{\mu}_k-\|A\|_2^2<c$, that is, $\overline{\mu}_k>\frac{\tau}{\|A\|_2^2+c}$. Let n_k denote the number of inner iterations in the kth outer iteration. For $0<\tau<1$, we obtain

$$\mu_{max}\tau^{n_k-1}\geqslant\mu_k\tau^{n_k-1}=\overline{\mu}_k>\tau/(\|A\|_2^2+c).$$

That is, $n_k\leqslant\lceil\frac{-\log(\mu_{max})-\log(\|A\|_2^2+c)}{\log(\tau)}+2\rceil$. □

Now, we are ready to give a convergence result about APGT methods.

Theorem 2.3. *Suppose $\mu_0\|A\|_2^2<1$ and let $\{x^k\}$ be the sequence generated by APGT-LS or APGT-C algorithm. Then, it holds that:*
(1) The sequence $\{x^k\}$ converges in a finite number of iterations.
(2) Let x^* be the accumulation point of $\{x^k\}$ and $I^*=\{i:x_i^*\neq0\}$. Then x^* is a local minimizer of the problem

$$\begin{aligned}&\min_x\|y-Ax\|_2^2\\ &s.t.\ 0<x_i\leqslant u,i\in I^*;\\ &\qquad x_i=0,i\notin I^*.\end{aligned} \tag{2.7}$$

Furthermore, if $|I^*|=s$, then x^* is a local minimizer of problem (2.1).

Proof. (1) From (2.4), the nonnegative sequence $\{\|y-Ax^k\|_2^2\}$ is nonincreasing and hence convergent. Due to its periodicity, the sequence must be eventually a constant. Furthermore, it follows from (2.4) that $p^{k+1}=x^k$ and $S^{k+1}=S^k$ when k is large enough. Thus the conclusion holds.

(2) According to (1), $p^{k+1}=x^k$ and $S^{k+1}=S^k$ with k large enough. Thus, we can know that $x^*=p^{k+1}=x^k$ and $Supp(x^k)=I^*$ for a large integer k. Since $x^{k+1}\in\arg\min_x\{\|y-Ax\|_2^2:0\leqslant x\leqslant u,Supp(x)\subset S^{k+1}\}$, x^{k+1} is also the minimizer of the problem

$$\begin{aligned}&\min_x\|y-Ax\|_2^2\\ &s.t.\ 0\leqslant x_i\leqslant u,i\in Supp(x^k);\\ &\qquad x_i=0,i\notin Supp(x^k).\end{aligned}$$

Thus, x^* is the minimizer of problem (2.7). Furthermore, if $|I^*| = s$, then x^* is a local minimizer of problem (2.1). In fact, let $U(x^*,\epsilon) = \{x \in \Omega_1 : \|x - x^*\| < \epsilon\}, U'(x^*,\epsilon) = \{x \in \Omega_1' : \|x - x^*\| < \epsilon\}$, where Ω_1 and Ω_1' are the feasible sets of problem (2.1) and (2.7), respectively, and $\epsilon = min\{x_i^* : i \in I^*\}$. Clearly, $\|y - Ax\|_2^2 \geqslant \|y - Ax^*\|_2^2$ for any $x \in U'(x^*,\epsilon)$. According to the definition of ϵ and the assumption that $|I^*| = s$, it is not hard to see that $U(x^*,\epsilon) = U'(x^*,\epsilon)$. This implies that x^* is a local minimizer of problem (2.1). □

2.2 Precise recovery of sparse vectors from accurate measurements

In this subsection, we present some results of precise recovery of sparse vectors from accurate measurements by our proposed APGT methods. Before proceeding, we give the restricted isometry constant [7],

$$\delta_s = \max_{|S| \leqslant s} \|A_S^* A_S - I\|_2,$$

where A_S denotes the submatrix of A obtained by keeping the columns indexed by S. This enables us to easily observe the following formula [7]:

$$|(w,(I - A^*A)t)| \leqslant \delta_s \|w\|_2 \|t\|_2, \qquad |Supp(w) \cup Supp(t)| \leqslant s.$$

Next we give the main precise recovery results for APGT method.

Theorem 2.4. *(1) For the APGT-LS algorithm, suppose that matrix $A \in R^{m\times n}$ is of full row rank and $cond(A) \leqslant 1.64$. Then for all $k \geqslant 0$, there exists $\mu_k \in [\mu_{min}, \mu_{max}]$, for any s-sparse solution $x \in \Omega_1$ of $y = Ax$, the sequence $\{x^k\}$ generated by the APGT-LS algorithm converges to x at a geometric rate ρ given by*

$$\|x^{k+1} - x\|_2 \leqslant \rho^k \|x^0 - x\|_2, \quad \rho := \sqrt{\frac{2(1-\mu_{min}\lambda_{min})^2}{1-(1-m\lambda_{min})^2}} < 1, \tag{2.8}$$

where $\lambda_{max} = \sigma_{max}^2, \lambda_{min} = \sigma_{min}^2, m = \frac{2}{\lambda_{min}+\lambda_{max}}$.

(2)For the APGT-C algorithm, suppose that the 3sth order restricted isometry constant of the measurement matrix $A \in R^{m\times n}$ satisfies $\delta_{3s} < 1/\sqrt{3}$. Then for any s-sparse solution $x \in \Omega_1$ of $y = Ax$, the sequence $\{x^k\}$ generated by the APGT-C algorithm converges to x at a geometric rate ρ_1 given by

$$\|x^{k+1} - x\| \leqslant \rho_1^k \|x^0 - x\|, \quad \rho_1 := \sqrt{\frac{2\delta_{3s}^2}{1-\delta_{2s}^2}} < 1. \tag{2.9}$$

Proof. (1) For the APGT-LS algorithm, we know from the first-order optimality conditions of (2.3) that

$$\langle z - x^{k+1}, A^*(Ax^{k+1} - y)\rangle \geqslant 0, \quad \forall z \text{ satisfies } 0 \leqslant z \leqslant u, Supp(z) \subseteq S^{k+1}.$$

Since $y = Ax$, it can be rewritten as

$$\langle z - x^{k+1}, A^*A(x^{k+1} - x)\rangle \geqslant 0.$$

Thus, we can derive

$$\begin{aligned} \|(x^{k+1} - x)_{S^{k+1}}\|_2^2 &= \langle (x - x^{k+1})_{S^{k+1}}, (x - x^{k+1})\rangle \\ &\leqslant \langle (x - x^{k+1})_{S^{k+1}}, (I - mA^*A)(x - x^{k+1})\rangle \\ &\leqslant \|I - mA^*A\|_2 \|x^{k+1} - x\|_2 \|(x^{k+1} - x)_{S^{k+1}}\|_2 \\ &= (1 - m\lambda_{min})\|x^{k+1} - x\|_2 \|(x^{k+1} - x)_{S^{k+1}}\|_2, \end{aligned} \tag{2.10}$$

where $m = 2/(\lambda_{min} + \lambda_{max})$. After simplification, we have $\|(x^{k+1} - x)_{S^{k+1}}\|_2 \leqslant (1 - m\lambda_{min})\|x^{k+1} - x\|_2$. Hence, we obtain

$$\begin{aligned} \|x^{k+1} - x\|_2^2 &= \|(x^{k+1} - x)_{S^{k+1}}\|_2^2 + \|(x^{k+1} - x)_{\overline{S^{k+1}}}\|_2^2 \\ &\leqslant (1 - m\lambda_{min})^2 \|x^{k+1} - x\|_2^2 + \|(x^{k+1} - x)_{\overline{S^{k+1}}}\|_2^2. \end{aligned}$$

Then we can obtain

$$\|x^{k+1}-x\|_2^2 \leqslant \frac{1}{1-(1-m\lambda_{min})^2}\|(x^{k+1}-x)_{\overline{S^{k+1}}}\|_2^2. \tag{2.11}$$

Let $S = Supp(x)$, we notice $\|(p^{k+1})_S\|_2^2 \leqslant \|(p^{k+1})_{S^{k+1}}\|_2^2$. Hence, by eliminating the contribution of $S \cap S^{k+1}$,

$$\|(p^{k+1})_{S\setminus S^{k+1}}\|_2^2 \leqslant \|(p^{k+1})_{S^{k+1}\setminus S}\|_2^2.$$

Further, we know that

$$\begin{aligned}\|(p^{k+1})_{S\setminus S^{k+1}}\|_2 &= \|(x-x^{k+1})_{\overline{S^{k+1}}} + (p^{k+1}-x)_{S\setminus S^{k+1}}\|_2\\ &\geqslant \|(x-x^{k+1})_{\overline{S^{k+1}}}\|_2 - \|(p^{k+1}-x)_{S\setminus S^{k+1}}\|_2.\end{aligned}$$

Thus, we obtain

$$\begin{aligned}\|(x-x^{k+1})_{\overline{S^{k+1}}}\|_2 &\leqslant \|(p^{k+1}-x)_{S\setminus S^{k+1}}\|_2 + \|(p^{k+1})_{S^{k+1}\setminus S}\|_2\\ &= \|(p^{k+1}-x)_{S\setminus S^{k+1}}\|_2 + \|(p^{k+1}-x)_{S^{k+1}\setminus S}\|_2\\ &\leqslant \sqrt{2}\|(p^{k+1}-x)_{S\cup S^{k+1}}\|_2.\end{aligned} \tag{2.12}$$

Next we need to find the relationship between $\|(p^{k+1}-x)_{S\cup S^{k+1}}\|_2$ and $\|x^k-x\|_2$. In fact,

$$\begin{aligned}\|(p^{k+1}-x)_{S\cup S^{k+1}}\|_2 &= \|(\Pi_{[0,u]}(S_{\mu_k}(x^k))-x)_{S\cup S^{k+1}}\|_2\\ &= \|(\Pi_{[0,u]}(x^k+\mu_k A^*(y-Ax^k)) - \Pi_{[0,u]}(x))_{S\cup S^{k+1}}\|_2\\ &\leqslant \|(x^k+\mu_k A^*(y-Ax^k)-x)_{S\cup S^{k+1}}\|_2\\ &= \|((I-\mu_k A^*A)(x^k-x))_{S\cup S^{k+1}}\|_2\\ &\leqslant \|I-\mu_k A^*A\|_2\|x^k-x\|_2\\ &= (1-\mu_k\lambda_{min})\|x^k-x\|_2.\end{aligned} \tag{2.13}$$

Combining (2.11), (2.12) and (2.13), we obtain

$$\|x^{k+1}-x\|_2 \leqslant \sqrt{\tfrac{2(1-\mu_k\lambda_{min})^2}{1-(1-m\lambda_{min})^2}}\|x^k-x\|_2 \leqslant \sqrt{\tfrac{2(1-\mu_{min}\lambda_{min})^2}{1-(1-m\lambda_{min})^2}}\|x^k-x\|_2.$$

Let $\frac{2(1-\mu_{min}\lambda_{min})^2}{1-(1-m\lambda_{min})^2} < 1$, then $\mu_{min} > (1-\sqrt{\frac{1-(1-m\lambda_{min})^2}{2}})/\lambda_{min}$. In addition, since $\mu_k < 1/\lambda_{max}$, we get $(1-\sqrt{\frac{1-(1-m\lambda_{min})^2}{2}})/\lambda_{min} < 1/\lambda_{max}$. Hence, we obtain $\lambda_{max}/\lambda_{min} \leqslant 2.691739$ which implies $cond(A) \leqslant 1.64$. Denote $\rho = \sqrt{\frac{2(1-\mu_{min}\lambda_{min})^2}{1-(1-m\lambda_{min})^2}}$ and $\rho < 1$. Thus, this completes the proof of the APGT-LS algorithm.

(2) As for the APGT-C algorithm, it follows from the second inequality in (2.10) and the definition of δ_{2s} that

$$\|(x^{k+1}-x)_{S^{k+1}}\|_2^2 \leqslant \delta_{2s}\|x^{k+1}-x\|_2\|(x^{k+1}-x)_{S^{k+1}}\|_2, \tag{2.14}$$

Similar to (2.11), we can obtain by (2.14)

$$\|x^{k+1}-x\|_2^2 \leqslant \frac{1}{1-\delta_{2s}^2}\|(x^{k+1}-x)_{\overline{S^{k+1}}}\|_2^2. \tag{2.15}$$

Also, similar to (2.13), we can prove that

$$\|(p^{k+1}-x)_{S\cup S^{k+1}\cup S^k}\|_2 \leqslant \delta_{3s}\|x^k-x\|_2. \tag{2.16}$$

Combining (2.12), (2.15) and (2.16), we arrive at

$$\|x^{k+1}-x\|_2 \leqslant \sqrt{\frac{2\delta_{3s}^2}{1-\delta_{2s}^2}}\|x^k-x\|_2.$$

The multiplicative coefficient $\rho_1 := \sqrt{2\delta_{3s}^2/(1-\delta_{2s}^2)}$ is less than one as soon as $2\delta_{3s}^2 < 1-\delta_{2s}^2$. Since $\delta_{2s} < \delta_{3s}$, this implies $\delta_{3s} < 1/\sqrt{3}$. Thus, we complete the proof. $\square$

Furthermore, we establish that the convergence of APGT methods requires finite iterations in precise recovery.

Corollary 2.5. *(1) Suppose that the matrix $A \in R^{m\times n}$ satisfies $cond(A) \leqslant 1.64$ and $\|x\|_0 = s$. Then any s-sparse vector $x \in \Omega_1$ is recovered by APGT-LS algorithm with $y = Ax$ in at most*

$$\lceil \frac{ln(\|x^0 - x\|_2/\xi)}{ln(1/\rho)} \rceil + 1$$

iterations, where ρ is defined in Theorem 2.4 and ξ is the smallest nonzero entry of x.

(2) Suppose that the matrix $A \in R^{m\times n}$ satisfies $\delta_{3s} < 1/\sqrt{3}$ and $\|x\|_0 = s$. Then any s-sparse vector $x \in \Omega_1$ is recovered by APGT-C algorithm with $y = Ax$ in at most

$$\lceil \frac{ln(\sqrt{2/3}\|x^0 - x\|_2/\xi)}{ln(1/\rho_1)} \rceil + 1$$

iterations, where ρ_1 is defined in Theorem 2.4 and agagin ξ is the smallest nonzero entry of x.

Proof. (1) If we find a sufficiently large integer k such that $S^k = S$, then the APGT-LS algorithm implies $x^k = x$. By the definition of S^k, this occurs if for all $i \in S, j \in \overline{S}$, we have

$$[\Pi_{[0,u]}(x^{k-1} + \mu_{k-1}A^*A(x - x^{k-1}))]_i \geqslant [\Pi_{[0,u]}(x^{k-1} + \mu_{k-1}A^*A(x - x^{k-1}))]_j. \tag{2.17}$$

For the left side term of the inequality (2.17), we have

$$\begin{aligned}
& [\Pi_{[0,u]}(x^{k-1} + \mu_{k-1}A^*A(x - x^{k-1}))]_i \\
\geqslant\ & [\Pi_{[0,u]}(x)]_i - [\Pi_{[0,u]}((I - \mu_{k-1}A^*A)(x^{k-1} - x))]_i \\
\geqslant\ & \xi - |((I - \mu_{k-1}A^*A)(x^{k-1} - x))]_i|.
\end{aligned}$$

For the right side term of the inequality (2.17), we notice that

$$\begin{aligned}
& [\Pi_{[0,u]}(x^{k-1} + \mu_{k-1}A^*A(x - x^{k-1}))]_j \\
=\ & [\Pi_{[0,u]}(x^{k-1} + \mu_{k-1}A^*A(x - x^{k-1}))]_j - x_j \\
=\ & [\Pi_{[0,u]}(x^{k-1} + \mu_{k-1}A^*A(x - x^{k-1}))]_j - [\Pi_{[0,u]}(x)]_j \\
\leqslant\ & |(x^{k-1} + \mu_{k-1}A^*A(x - x^{k-1}))_j - x_j| \\
=\ & |((I - \mu_{k-1}A^*A)(x^{k-1} - x))_j|.
\end{aligned}$$

Thus, it is not hard to verify that (2.17) holds when

$$\xi - |((I - \mu_{k-1}A^*A)(x^{k-1} - x))]_i| \geqslant |((I - \mu_{k-1}A^*A)(x^{k-1} - x))_j|.$$

In addition, we notice that

$$\begin{aligned}
& |((I - \mu_{k-1}A^*A)(x^{k-1} - x))_i| + |((I - \mu_{k-1}A^*A)(x^{k-1} - x))_j| \\
\leqslant\ & \sqrt{2}\|((I - \mu_{k-1}A^*A)(x^{k-1} - x))_{\{j,l\}}\|_2 \\
\leqslant\ & \sqrt{2}\|I - \mu_{k-1}A^*A\|_2\|x^{k-1} - x\|_2 \\
\leqslant\ & \sqrt{2}(1 - \mu_{min}\lambda_{min})\|x^{k-1} - x\|_2 \\
=\ & \sqrt{1 - (1 - m\lambda_{min})^2}\rho\|x^{k-1} - x\|_2 \\
\leqslant\ & \rho^{k-1}\|x^0 - x\|_2.
\end{aligned}$$

Thus, (2.17) holds when $\xi \geqslant \rho^{k-1}\|x^0 - x\|_2$. This is true provided that $k \geqslant \lceil \frac{ln(\|x^0 - x\|_2/\xi)}{ln(1/\rho)} \rceil + 1$.

(2)As for the APGT-C algorithm, the inequality (2.17) can be written as

$$[\Pi_{[0,u]}(x^{k-1} + A^*A(x - x^{k-1}))]_i \geqslant [\Pi_{[0,u]}(x^{k-1} + A^*A(x - x^{k-1}))]_j. \tag{2.18}$$

Similarly, (2.18) holds when

$$\xi - |((I - A^*A)(x^{k-1} - x))]_i| \geqslant |((I - A^*A)(x^{k-1} - x))_j|.$$

Hence,

$$\begin{aligned} & |((I - A^*A)(x^{k-1} - x))_i| + |((I - A^*A)(x^{k-1} - x))_j| \\ \leqslant\ & \sqrt{2}\|((I - A^*A)(x^{k-1} - x))_{\{j,l\}}\|_2 \\ \leqslant\ & \sqrt{2}\delta_{3s}\|x^{k-1} - x\|_2 \\ =\ & \sqrt{1 - \delta_{2s}^2}\rho_1\|x^{k-1} - x\|_2 \\ \leqslant\ & \sqrt{2/3}\rho_1^{k-1}\|x^0 - x\|_2. \end{aligned}$$

Therefore, (2.18) holds when $\xi \geqslant \sqrt{2/3}\rho_1^{k-1}\|x^0 - x\|_2$. Thus, the required smallest of integer k is given as in the conclusion. □

2.3 Approximate recovery of vectors from flawed measurements

We extend the previous conclusion of the *APGT* methods to the case of approximate recovery of vectors from flawed measurements. Before proceeding, we give the following observation in [7], for any $e \in R^m$,

$$\|(A^*e)_S\|_2 \leqslant \sqrt{1 + \delta_s}\|e\|_2, \qquad \text{whenever } |S| \leqslant s.$$

Theorem 2.6. *(1) Suppose that matrix $A \in R^{m\times n}$ is of full row rank, satisfying $cond(A) \leqslant 1.64$. Then for any $x \in R^n$ and $e \in R^m$, if S denotes the index set of s largest entries of x, the sequence $\{x^k\}$ generated by the APGT-LS algorithm with $y = Ax + e$ satisfies*

$$\|x^{k+1} - x_S\|_2 \leqslant \rho^k\|x^0 - x_S\|_2 + \tau\frac{1 - \rho^k}{1 - \rho}\|Ax_{\overline{S}} + e\|_2, \quad \forall k \geqslant 0, \tag{2.19}$$

where ρ is given in (2.8) and

$$\tau = \frac{\mu_{max}\sqrt{2(1 + \delta_{2s})}}{\sqrt{1 - (1 - m\lambda_{min})^2}} + \frac{\sqrt{1 + \delta_s}}{\lambda_{min}}.$$

(2) In particular, suppose the $3s$-th restricted isometry constant of the measurement matrix $A \in R^{m\times n}$ satisfies $\delta_{3s} \leqslant 1/\sqrt{3} \approx 0.57735$. Then for any $x \in R^n$ and $e \in R^m$, if S denotes the index set of s largest entries of x, the sequence $\{x^k\}$ generated by the APGT-C algorithm with $y = Ax + e$ satisfies

$$\|x^{k+1} - x_S\|_2 \leqslant \rho_1^k\|x^0 - x_S\|_2 + \tau_1\frac{1 - \rho_1^k}{1 - \rho_1}\|Ax_{\overline{S}} + e\|_2, \quad \forall k \geqslant 0, \tag{2.20}$$

where ρ_1 is given in (2.9) and

$$\tau_1 := \frac{\sqrt{2(1 - \delta_{2s})} + \sqrt{1 + \delta_s}}{1 - \delta_{2s}} \leqslant 5.15.$$

Proof. (1) According to the first-order optimality conditions of (2.3), we have

$$\langle z - x^{k+1}, A^*(Ax^{k+1} - y)\rangle \geqslant 0, \quad \forall z \text{ satisfies } 0 \leqslant z \leqslant u, Supp(z) \subseteq S^{k+1}.$$

The above inequality can be rewritten

$$\langle z - x^{k+1}, A^*A(x^{k+1} - x_S)\rangle + \langle x^{k+1} - z, A^*e'\rangle \geqslant 0$$

by using $e' = Ax_{\overline{S}} + e$ [7]. Hence, we get

$$\begin{aligned} & \|(x^{k+1} - x_S)_{S^{k+1}}\|_2^2 = \langle (x_S - x^{k+1})_{S^{k+1}}, x_S - x^{k+1}\rangle \\ \leqslant\ & \langle (x_S - x^{k+1})_{S^{k+1}}, (I - mA^*A)(x_S - x^{k+1})\rangle + m\langle e', A((x^{k+1} - x_S)_{S^{k+1}})\rangle \\ \leqslant\ & \|I - mA^*A\|_2\|x^{k+1} - x_S\|_2\|(x^{k+1} - x_S)_{S^{k+1}}\|_2 + m\sqrt{1 + \delta_s}\|e'\|_2\|(x^{k+1} - x_S)_{S^{k+1}}\|_2 \\ =\ & (1 - m\lambda_{min})\|x^{k+1} - x_S\|_2\|(x^{k+1} - x_S)_{S^{k+1}}\|_2 + m\sqrt{1 + \delta_s}\|e'\|_2\|(x^{k+1} - x_S)_{S^{k+1}}\|_2, \end{aligned} \tag{2.21}$$

where $m = \frac{2}{\lambda_{min}+\lambda_{max}}$. After simplification, we obtain

$$\|(x^{k+1} - x_S)_{S^{k+1}}\|_2 \leqslant (1 - m\lambda_{min})\|x^{k+1} - x_S\|_2 + m\sqrt{1+\delta_s}\|e'\|_2.$$

Thus,

$$\begin{aligned} & \|x^{k+1} - x_S\|_2^2 \\ = \; & \|(x^{k+1} - x_S)_{S^{k+1}}\|_2^2 + \|(x^{k+1} - x_S)_{\overline{S^{k+1}}}\|_2^2 \\ \leqslant \; & \|(x^{k+1} - x_S)_{\overline{S^{k+1}}}\|_2^2 + ((1 - m\lambda_{min})\|x^{k+1} - x_S\|_2 + m\sqrt{1+\delta_s}\|e'\|_2)^2, \end{aligned}$$

which implies $P(\|x^{k+1} - x_S\|_2) \leqslant 0$ for the following quadratic polynomial

$$\begin{aligned} P(t) = \; & (1 - (1 - m\lambda_{min})^2)t^2 - 2m\sqrt{1+\delta_s}(1 - m\lambda_{min})\|e'\|_2 t \\ & -(\|(x^{k+1} - x_S)_{\overline{S^{k+1}}}\|_2^2 + m^2(1+\delta_s)\|e'\|_2^2). \end{aligned}$$

Since $(1 - (1 - m\lambda_{min})^2) > 0$, we know that $\|x^{k+1} - x_S\|_2$ is bounded by the largest root of $P(t) = 0$, that is,

$$\begin{aligned} & \|x^{k+1} - x_S\|_2 \\ \leqslant \; & \frac{m\sqrt{1+\delta_s}(1-m\lambda_{min})\|e'\|_2}{1-(1-m\lambda_{min})^2} + \frac{\sqrt{(1-(1-m\lambda_{min})^2)\|(x^{k+1}-x_S)_{\overline{S^{k+1}}}\|_2^2 + m^2(1+\delta_s)\|e'\|_2^2}}{1-(1-m\lambda_{min})^2} \\ \leqslant \; & \frac{m\sqrt{1+\delta_s}(1-m\lambda_{min})\|e'\|_2}{1-(1-m\lambda_{min})^2} + \frac{\sqrt{(1-(1-m\lambda_{min})^2)}\|(x^{k+1}-x_S)_{\overline{S^{k+1}}}\|_2 + m\sqrt{(1+\delta_s)}\|e'\|_2}{1-(1-m\lambda_{min})^2} \\ = \; & \frac{1}{\sqrt{1-(1-m\lambda_{min})^2}}\|(x^{k+1} - x_S)_{\overline{S^{k+1}}}\|_2 + \frac{m\sqrt{1+\delta_s}}{1-(1-m\lambda_{min})}\|e'\|_2 \\ = \; & \frac{1}{\sqrt{1-(1-m\lambda_{min})^2}}\|(x^{k+1} - x_S)_{\overline{S^{k+1}}}\|_2 + \frac{\sqrt{1+\delta_s}}{\lambda_{min}}\|e'\|_2. \end{aligned} \tag{2.22}$$

Furthermore, $\|(p^{k+1})_{S\setminus S^{k+1}}\|_2 \leqslant \|(p^{k+1})_{S^{k+1}\setminus S}\|_2$ still holds here, that is,

$$\|(\Pi_{[0,u]}(x^k + \mu_k A^*(y - Ax^k)))_{S\setminus S^{k+1}}\|_2 \leqslant \|(\Pi_{[0,u]}(x^k + \mu_k A^*(y - Ax^k)))_{S^{k+1}\setminus S}\|_2.$$

For the right side of this inequality, we have

$$\begin{aligned} & \|(\Pi_{[0,u]}(x^k + \mu_k A^*(y - Ax^k)))_{S^{k+1}\setminus S}\|_2 \\ = \; & \|(\Pi_{[0,u]}(x^k + \mu_k A^*(Ax_S + e' - Ax^k)))_{S^{k+1}\setminus S}\|_2 \\ = \; & \|(\Pi_{[0,u]}((I - \mu_k A^*A)(x^k - x_S) + x_S + \mu_k A^*e'))_{S^{k+1}\setminus S}\|_2 \\ \leqslant \; & \|(\Pi_{[0,u]}((I - \mu_k A^*A)(x^k - x_S))_{S^{k+1}\setminus S}\|_2 + \|(\Pi_{[0,u]}(x_S + \mu_k A^*e'))_{S^{k+1}\setminus S}\|_2 \\ \leqslant \; & \|((I - \mu_k A^*A)(x^k - x_S))_{S^{k+1}\setminus S}\|_2 + \|(\mu_k A^*e')_{S^{k+1}\setminus S}\|_2. \end{aligned}$$

For the left side of this inequality, we have

$$\begin{aligned} & \|(\Pi_{[0,u]}(x^k + \mu_k A^*(y - Ax^k)))_{S\setminus S^{k+1}}\|_2 \\ = \; & \|(\Pi_{[0,u]}(x_S + x^k + \mu_k A^*(y - Ax^k) - x_S))_{S\setminus S^{k+1}}\|_2 \\ = \; & \|(\Pi_{[0,u]}((I - \mu_k A^*A)(x^k - x_S) + x_S + \mu_k A^*e'))_{S\setminus S^{k+1}}\|_2 \\ = \; & \|(\Pi_{[0,u]}((I - \mu_k A^*A)(x^k - x_S) + x_S - x^{k+1} + \mu_k A^*e'))_{S\setminus S^{k+1}}\|_2 \\ \geqslant \; & \|(x_S - x^{k+1})_{\overline{S^{k+1}}}\|_2 - \|(\Pi_{[0,u]}((I - \mu_k A^*A)(x^k - x_S))_{S\setminus S^{k+1}}\|_2 - \|(\Pi_{[0,u]}(\mu_k A^*e'))_{S\setminus S^{k+1}}\|_2 \\ \geqslant \; & \|(x_S - x^{k+1})_{\overline{S^{k+1}}}\|_2 - \|((I - \mu_k A^*A)(x^k - x_S))_{S\setminus S^{k+1}}\|_2 - \|(\mu_k A^*e')_{S\setminus S^{k+1}}\|_2. \end{aligned}$$

Combining these inequalities, it is not hard to know that

$$\begin{aligned} \|(x_S - x^{k+1})_{\overline{S^{k+1}}}\|_2 & \leqslant \|((I - \mu_k A^*A)(x^k - x_S))_{S^{k+1}\setminus S}\|_2 + \|(\mu_k A^*e')_{S^{k+1}\setminus S}\|_2 \\ & \quad + \|((I - \mu_k A^*A)(x^k - x_S))_{S\setminus S^{k+1}}\|_2 + \|(\mu_k A^*e')_{S\setminus S^{k+1}}\|_2 \\ & \leqslant \sqrt{2}(\|((I - \mu_k A^*A)(x^k - x_S))_{S\cup S^{k+1}}\|_2 + \|(\mu_k A^*e')_{S\cup S^{k+1}}\|_2) \\ & \leqslant \sqrt{2}(\|I - \mu_k A^*A\|_2\|x^k - x_S\|_2 + \mu_k\sqrt{1+\delta_{2s}}\|e'\|_2) \\ & \leqslant \sqrt{2}((1 - \mu_{min}\lambda_{min})\|x^k - x_S\|_2 + \mu_{max}\sqrt{1+\delta_{2s}}\|e'\|_2). \end{aligned} \tag{2.23}$$

Substituting (2.23) into (2.22), we obtain

$$\|x^{k+1} - x_S\|_2 \leqslant \rho\|x^k - x_S\|_2 + \tau\|e'\|_2 \quad \forall k \geqslant 0.$$

Since $cond(A) \leqslant 1.64$, we know from the proof of Theorem 2.4 that $\rho < 1$. Thus the estimate (2.16) holds, and the proof is completed.

(2) Specially, for APGT-C algorithm, (2.21) with $m = 1$ can be further written as

$$\|(x^{k+1} - x_S)_{S^{k+1}}\|_2^2 \leqslant \delta_{2s}\|x^{k+1} - x_S\|_2\|(x^{k+1} - x_S)_{S^{k+1}}\|_2 + \sqrt{1+\delta_s}\|e'\|_2\|(x^{k+1} - x_S)_{S^{k+1}}\|_2.$$

After simplification, we obtain $\|(x^{k+1} - x_S)_{S^{k+1}}\|_2 \leqslant \delta_{2s}\|x^{k+1} - x_S\|_2 + \sqrt{1+\delta_s}\|e'\|_2$. With the same deduction, it is not hard to get the upper-bound of $\|x^{k+1} - x_S\|_2$, that is,

$$\|x^{k+1} - x_S\|_2 \leqslant \frac{1}{\sqrt{1-\delta_{2s}^2}}\|(x^{k+1} - x_S)_{\overline{S^{k+1}}}\|_2 + \frac{\sqrt{1+\delta_s}}{1-\delta_{2s}}\|e'\|_2. \tag{2.24}$$

Let $\mu_k = 1$, (2.23) can be expressed by

$$\|(x_S - x^{k+1})_{\overline{S^{k+1}}}\|_2 \leqslant \sqrt{2}(\delta_{3s}\|x^k - x_S\|_2 + \sqrt{1+\delta_{2s}}\|e'\|_2). \tag{2.25}$$

Combining (2.24) and (2.25), it yields

$$\|x^{k+1} - x_S\|_2 \leqslant \sqrt{\frac{2\delta_{3s}^2}{1-\delta_{2s}^2}}\|x^k - x_S\|_2 + \frac{\sqrt{2(1-\delta_{2s})} + \sqrt{1+\delta_s}}{1-\delta_{2s}}\|e'\|_2.$$

Since $\delta_{3s} < 1/\sqrt{3}$, $\rho_1 := \sqrt{2\delta_{3s}^2/(1-\delta_{2s}^2)} < 1$. Thus, (2.20) holds. □

3 l_0 sparse minimization problems with equality and box constraints

In this section, we consider the l_0 sparse optimization problem with equality and box constraints

$$\begin{aligned} &\min \|y - Ax\|_2^2 \\ &s.t. \ \ x \in \Omega_2, \end{aligned} \tag{3.1}$$

where $\Omega_2 = \{x \in R^n : e^T x = d, 0 \leqslant x \leqslant u, \|x\|_0 \leqslant s\}$.

We also apply the APGT methods to solving (3.1), mainly including two parts: the first one is to determine the iterative index sets by minimizing a surrogate function; the second one is to solve the box and equality constrained least squares subproblem on the determined indices sets. We can prove that the solution of subproblem

$$p^{k+1} \in \min_{x\in\Omega_2} \|x - S_{\mu_k}(x^k)\|_2^2 \tag{3.2}$$

also has the closed form, and the optimal index set can be determined by Theorem 3.1.

Theorem 3.1. *The subproblem* $\min\limits_{x\in\Omega_2} \|x - S_{\mu_k}(x^k)\|_2^2$ *has the closed-form solution*

$$p_i^{k+1} = \begin{cases} \Pi_{[0,u_i]}([S_{\mu_k}(x^k)]_i + \lambda), & i \in I_s(S_{\mu_k}(x^k)); \\ 0, & otherwise \end{cases}$$

where λ *satisfies*

$$\sum_{i\in I_s(S_{\mu_k}(x^k))} \Pi_{[0,u_i]}([S_{\mu_k}(x^k)]_i + \lambda) = d. \tag{3.3}$$

Proof. Consider the Lagrangian function of the objective function of the subproblem (3.2)

$$f(x,\lambda) = \|x - S_{\mu_k}(x^k)\|_2^2 - 2\lambda(e^T x - d),$$

which can be simplified by

$$\arg\min_x f(x,\lambda) = \arg\min_x \|x - S_{\mu_k}(x^k) - \lambda e\|_2^2 \tag{3.4}$$

In Lemma 2.1, the optimal index set of solution for problem (3.4) is $I_s(S_{\mu_k}(x^k) + \lambda e)$. Since λ is a constant, $I_s(S_{\mu_k}(x^k) + \lambda e) = I_s(S_{\mu_k}(x^k))$. Hence, we can give the optimal solution of this problem as long as there exists one λ satisfying the equation (3.3). In fact, such a λ exists as proved in [26], thus the proof is completed. □

Theorem 3.1 implies that the equality constraint $e^T x = d$ has no influence on the determination of S^{k+1}. Then we give the main steps of APGT methods:

(1) Let $S^{k+1} = I_s(S_{\mu_k}(x^k)$, and solve $p^{k+1} \in \arg\min_{x\in\Omega_2} \|x - S_{\mu_k}(x^k)\|_2^2$;

(2) Solve $x^{k+1} \in \arg\min_x\{\|y - Ax\|_2^2 : e^T x = d, 0 \leqslant x \leqslant u, Supp(x) \subset S^{k+1}\}$.

Remarks

(i) If $\|y - Ax^{k+1}\|_2^2 - \|y - Ax^k\|_2^2 \leqslant -c\|p^{k+1} - x^k\|_2^2$ holds with updated μ_k in each iteration, then the method is APGT-LS algorithm;

(ii) If $\mu_k = 1$ for each iteration, then it is APGT-C algorithm;

(iii) The problem in Step (2) is a quadratic programming with box and equality constraints, which can be solved in Matlab by `quadprog`;

(iv) The sequence $\{x^k\}$ generated by APGT methods for (3.1) converges in a finite number of iteration. Moreover, the theory and results in signal recovery still hold for (3.1).

4 Numerical results

In this section, we present several numerical experiments to demonstrate the high performance of APGT methods by applying it to compressive sensing with box constraint (signal recovery) and general index tracking. All the computational tests were conducted on a HP dx7408 PC (Intel core E4500 CPU, 2.2GHz,1GB RAM)with using Matlab 7.9 (R2009b).

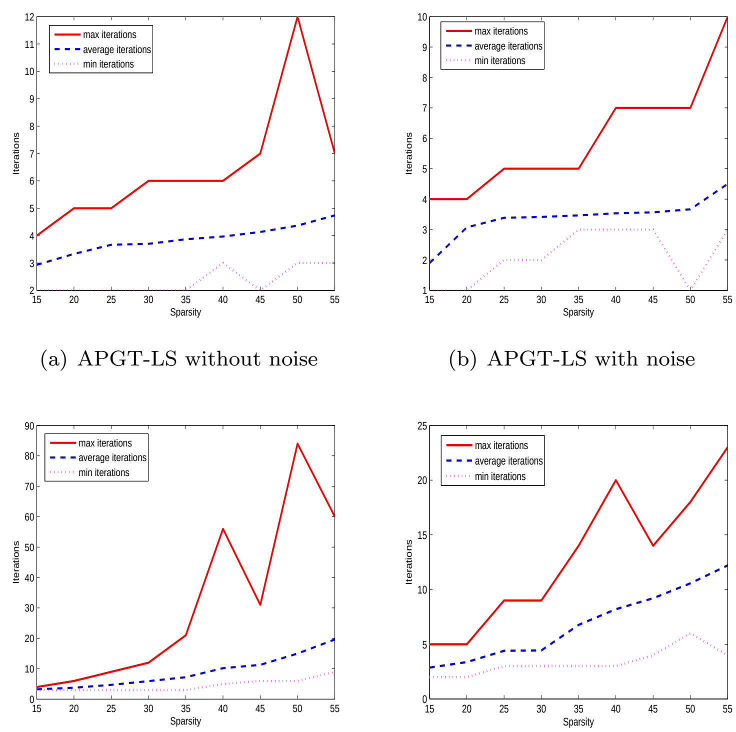
4.1 Application in compressive sensing

In this subsection, we make some experiments by applying the APGT methods to the signal recovery. These empirical tests aim at verifying the finite number of iterations, the geometric convergence, effectiveness and robustness of the APGT methods. We consider the general signal recovery problem with box constraints, that is, problem (2.1). Here, $A \in R^{m\times n}$ is the dictionary matrix, $y \in R^m$ is the measurement vector, and $x \in R^n$ is the signal we would like to recover.

In each test, the dictionary matrix A is generated by the Gaussian distribution suggested in [1] and the measurement vector is generated by product of the matrix A and a random real-valued vector x with sparsity s. The length of the signal is $n = 512$ and the noise added is the white noise $\varepsilon \in N(0, \sigma^2)$ with $\sigma = 0.1$ in the noisy case. The performance of algorithm is measured by how few measurements are required to exactly recover a signal. The fewer the measurements used by an algorithm, the better it is. As for the parameters in APGT methods, we set $\mu_{max} = 1/\|A\|_2^2, \mu_{min} = 10^{-16}, \tau = 0.5, c = 10^{-4}, u = 0.5$.

1) The number of iterations.

Theorem 2.3 shows that the convergence of the APGT methods requires only a finite number of iterations. Hence, the natural stop criterion is $S^{k+1} = S^k$, and this is incorporated in our experiments. We set $m = 330$, and the sparsity s ranges from 15 to 55 with step 5. For a fixed sparsity s, we conduct experiments 100 times randomly, and record the maximum iterations, minimum iterations and average iterations. The results are presented in Figure 1. In Figure 1, the x-axis denotes the sparsity and the y-axis denotes the number of iterations.



(a) APGT-LS without noise (b) APGT-LS with noise

(c) APGT-C without noise (d) APGT-C with noise

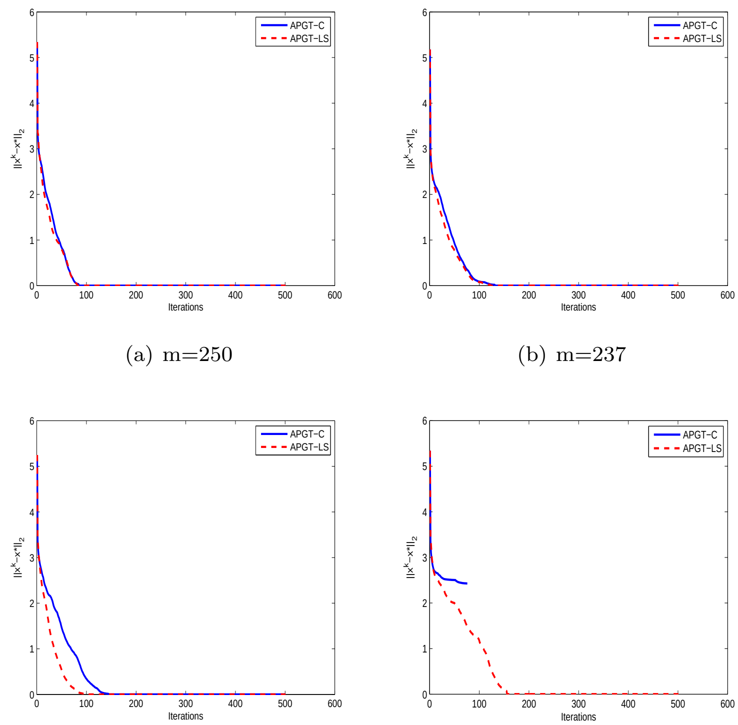
Figure 1 The iterations of APGT methods.

From Figure 1, we can find that (i) For the APGT-LS algorithm, the maximum number of iterations doesn't exceed 12 when without noise and 10 when with noise; (ii) For the APGT-C algorithm, the maximum number of iterations doesn't exceed 90 when without noise and 25 when with noise. Moreover, we also notice that the average number of iterations increase as the sparsity increases.

2) Geometric rate of convergence of APGT methods in exact recovery.
As in Theorem 2.4, we have proved that the sequence generated by the APGT methods converges to a s-sparse solution x satisfying $y = Ax$ at the geometric speed. Thus, we conduct experiments to investigate its convergence speed by applying the algorithm to signal recovery without noise. Here, the sparsity of signal x is $s = 130$ and the number of measurements m ranges in $[0, 512]$. The experiment is stopped when the number of iterations exceeds 500 or $S^{k+1} = S^k$. Some of results are given in Figure 2. For each subfigure in Figure 2, the x-axis denotes the numbers of iterations, and the y-axis denotes the Euclidean distance between the iteration point x^k and the optimal solution x.

From Figure 2, it is clear to notice that the APGT methods converge with a geometric speed. With enough measurements ($m = 250$), the APGT-LS algorithm and the APGT-C algorithm almost have the same performance. With fewer measurements, the APGT-LS algorithm has a faster convergence speed than the APGT-C algorithm, for example $m = 235$. Moreover, we can find that with few measurements ($m = 233$), the APGT-C algorithm fails to recover the signal while the APGT-LS algorithm succeed. All these indicate the choice of parameter μ can (i) accelerate the convergence speed in some extent; (ii) improve the effectiveness of APGT methods when taking not too much measurements.

3) Effectiveness and robustness of the APGT methods.
Consider the signal recovery under noiseless and noisy conditions respectively. The sparsity of the signal is $s = 130$ and the number of measurements is m. The stop criterion is also the number of iterations exceeds 500 or $S^{k+1} = S^k$. The mean square error (MSE) between the recovered signal and the original signal is computed, and the CPU time in seconds for running the algorithm is also recorded. Experiment



(a) m=250 (b) m=237

(c) m=235 (d) m=233

Figure 2 The geometric convergence of APGT methods.

results are provided in Tables 1 and 2.

Table 1 Recovery results of variable measurements without noise.

m	Method	MSE	time	m	Method	MSE	time
330	APGT-LS	1.06e-15	14.86	250	APGT-LS	3.46e-15	13.60
	APGT-C	8.28e-16	167.01		APGT-C	9.44e-16	183.89
239	APGT-LS	3.70e-15	26.84	237	APGT-LS	1.96e-15	21.52
	APGT-C	1.08e-15	157.89		APGT-C	8.68e-16	173.59
235	APGT-LS	2.45e-15	19.29	234	APGT-LS	4.34e-15	25.69
	APGT-C	1.05e-15	162.97		APGT-C	1.18e-15	133.05
233	APGT-LS	3.13e-15	29.69	232	APGT-LS	2.52	9.826
	APGT-C	2.43	9.79		APGT-C	2.78	11.05

From Table 1, we can find that both the APGT-LS algorithm and the APGT-C algorithm can accurately recover the signal when $m \geqslant 234$. In this case, APGT-C algorithm attain the higher accuracy but with much more running time than APGT-LS algorithm. When $m = 233$, APGT-LS algorithm succeed in the recovery of signal while APGT-C algorithm fails, which indicates that $m = 233$ is the phase transition point for APGT-C algorithm. Moreover, both the APGT-LS algorithm and the APGT-C algorithm fail in the signal recovery when $m = 232$, implying $m = 232$ is the phase transition point for the APGT-LS algorithm. All these also show that the choice of parameter μ can improve the effectiveness of the APGT methods with few samplings.

In order to understand the effect of noise better, we have used the Oracle [19] to examine the recovery capability of the algorithms in the experiments. For each algorithm, we have calculated the ratio of the MSE generated from the algorithm and the Oracle, listed as "Ratio" in Table 2. Hence, the more close the ratio is to 1, the better the algorithm, and the stronger the robustness of the algorithm correspondingly.

Table 2 Recovery results of variable measurements with noise.

m	Method	MSE	Ratio	time	m	Method	MSE	Ratio	time
330	APGT-LS	2.58	1.03	6.33	300	APGT-LS	2.98	1.03	6.65
	APGT-C	2.56	1.02	5.93		APGT-C	2.99	1.03	5.72
	Oracle	2.53				Oracle	2.95		
275	APGT-LS	3.36	0.98	7.08	250	APGT-LS	4.17	1.01	5.62
	APGT-C	3.44	1.00	5.37		APGT-C	4.19	1.02	5.09
	Oracle	3.43				Oracle	4.11		
249	APGT-LS	3.77	0.89	5.20	232	APGT-LS	3.98	0.82	4.94
	APGT-C	3.68	0.87	5.20		APGT-C	4.04	0.83	7.88
	Oracle	4.30				Oracle	4.85		
231	APGT-LS	4.02	0.82	7.14	230	APGT-LS	3.89	0.79	8.53
	APGT-C	3.98	0.81	9.33		APGT-C	3.86	0.78	8.04
	Oracle	4.94				Oracle	4.94		

From Table 2, we find that $m = 249$ is the phase transition point for both the APGT-LS algorithm and the APGT-C algorithm since the Ratios of the two algorithms change dramatically. Moreover, if we regard an algorithm to have failed in the exact recovery when its ratio is less than 0.9 or more than 1.1, by observing the Ratio values in Table 2, we find the APGT-LS algorithm and the APGT-C algorithm both yield highly accurate recovery results when $m \geqslant 250$; both the two algorithms fail in signal recovery when $m \leqslant 249$. This shows that the APGT methods in particular provides very nice signal recovery with noise before the phase transition value is reached.

4.2 Application in index tracking

In this subsection, we apply the APGT-LS algorithm to index tracking according to its excellent performance in compressed sensing experiments. Index tracking aims at replicating the performance and risk profile of a given market index, and constructs a sparse tracking portfolio such that the performance of the portfolio is as close as possible to that of the market index. Thus, we can propose the following general index tracking model:

$$\begin{array}{ll} \min\limits_{x} & TE(x) = \frac{1}{T}\|y - Rx\|_2^2 \\ \text{s.t.} & e^T x = 1 \\ & \|x\|_0 \leqslant s \\ & 0 \leqslant x_i \leqslant u, \end{array} \tag{4.1}$$

where $x \in R^n$ is the weight vector of n index constituents, y is the $T \times 1$ vector of index returns, and R is a $T \times n$ matrix of the index constituents returns. It is not hard to notice (4.1) is a special case of problem(3.1).

There have been many works on index tracking (See [2, 27–29]). We mainly conduct two experiments to compare the performance of the APGT-LS algorithm applied to (4.1) with two other algorithms, that is, the hybrid evolutionary algorithm in [29] and the hybrid half thresholding algorithm in [2]. For convenience of presentation, we abbreviate other two approaches as MIP and $l_{1/2}$ since they are the methods for mixed integer programming and $l_{1/2}$ models, respectively.

The data sets used in our experiments are selected from the standard ones in OR-library [30] and the CSI 300 index from China Shanghai-Shenzhen stock market. For the standard data sets, weekly prices of the stocks from 1992 to 1997 of Hang Seng (Hong Kong), DAX 100 (Germany), FTSE (Great Britain), Standard and Poor's 100 (USA), the Nikkei index (Japan), the Standard and Poor's 500 (USA), Russell 2000 (USA) and Russell 3000 (USA) are used. For CSI 300 index, the daily prices of 300 stocks from 2011 to 2013 in China stock market are considered. According to the sample scale, we divide the above data sets into two categories: small data sets including Hang Seng, DAX 100, FTSE , Standard and Poor's 100, the Nikkei index; and large data sets including CSI 300, Standard and Poor's 500, Russell 2000 and Russell 3000. The tackling of each data set is the same as in Torrubiano and Alberto [29].

For the APGT-LS algorithm, the parameter settings are the same with signal recovery. For the hybrid evolutionary algorithm, we set the lower bound to 0.01, the upper bound to 0.5, initial population size to 100, mutation probability to 1%, cross probability to 30%. For the hybrid half thresholding algorithm, the lower and upper bounds are chosen to be 0.01 and 0.5, respectively. We terminate the above three algorithms when the absolute error of the function values over two consecutive iterations is below 10^{-8}, or the maximum iteration is 1000.

We measure the performance of each algorithm by the following two criteria as in [2]. Before proceeding, denote TEI_A and TEO_A the in-sample errors and out-of-sample errors of a portfolio generated by the method A respectively.

(i) Consistency: $Cons(A) = |TEI_A - TEO_A|$;

(ii) Superiority of out-of-sample: $SupO(A, B) = \frac{TEO_B - TEO_A}{TEO_B} \times 100\%$.

We present numerical results in Tables 3 to 6, where N denotes the number of samples contained in a data set. In particular, we report in Tables 3 and 5 in-sample error and out-of sample error of the portfolios generated by the aforementioned three methods. In Table 4, we report the consistency between in-sample and out-of-sample errors, and the superiority of out-of-sample errors for the portfolios generated by these methods. In Table 6, we present the CPU time and superiority of out-of-sample errors of the portfolios given by these methods.

From Table 4, we can make the following observations:
(i) The APGT-LS algorithm generally has higher consistency between in-sample error and out-of-sample error than the MIP- and $l_{1/2}$-based methods (namely, hybrid evolutionary and half thresholding algorithms) since $Cons(APGT\text{-}LS) < Cons(MIP)$ holds for 97% (29/30) instances and $Cons(APGT\text{-}LS) < Cons(l_{1/2})$ holds for 83% (25/30) instances;
(ii) The APGT-LS algorithm is generally superior to the MIP- and $l_{1/2}$-based methods in terms of out-of-sample error since $SupO(APGT\text{-}LS, MIP) > 0$ holds for 90% (27/30) instances and $SupO(APGT\text{-}LS, l_{1/2}) > 0$ holds for 97% (29/30) instances.

In addition, we can have the following observations from Table 6.
(i) The APGT-LS algorithm is generally superior to the MIP- and $l_{1/2}$-based methods in terms of out-of-sample error since $SupO(APGT\text{-}LS, MIP) > 0$ holds in 92.9%(26/28) cases and $SupO(APGT\text{-}LS, l_{1/2}) > 0$ holds in 92.9%(26/28) cases;
(ii) The APGT-LS algorithm also generally outperforms the MIP- and $l_{1/2}$-based methods in terms of speed.

5 Concluding remarks

In this paper we have proposed an efficient adaptive projected gradient thresholding method for solving box-constrained l_0 problems and for solving l_0 problems with equality and box constraints, respectively. At each iteration, we have showed that each subproblem has a closed-form solution, which can be computed in a linear time. Under some suitable assumptions, we have showed that any accumulation point of the sequence generated by the APGT methods is a local minimizer of the two kinds of l_0 problem. For signal recovery problem, the sequence generated by APGT methods converges to the accurate sparse solution with geometric speed and we can recover the sparse signal in finite number of iterations. We have also conducted a series of empirical tests to test the performance of APGT methods. Firstly, we applied the algorithm to signal recovery with box-bounded constraints to verify the finite iterations, geometric convergence speed and effectiveness and robustness of APGT methods. The empirical results show that the theories aforementioned are all verified. Secondly, we conducted numerical experiments on the data sets from OR-library [30] and the CSI 300 index from China Shanghai-Shenzhen stock market

Table 3 The in-sample and out-of-sample tracking errors on small data sets.

Index	Sparsity	APGT-LS		MIP		$l_{1/2}$	
	s	TEI	TEO	TEI	TEO	TEI	TEO
Hang	5	6.07e-5	5.66e-5	4.71e-5	7.19e-5	6.60e-5	5.81e-5
Seng	6	6.88e-5	3.93e-5	4.13e-5	5.44e-5	4.97e-5	3.76e-5
(N=31)	7	2.75e-5	3.22e-5	3.27e-5	5.37e-5	2.89e-5	3.73e-5
	8	3.64e-5	3.13e-5	2.50e-5	4.41e-5	2.80e-5	3.38e-5
	9	3.28e-5	2.70e-5	2.08e-5	3.30e-5	2.06e-5	3.08e-5
	10	2.34e-5	1.73e-5	1.81e-5	2.73e-5	1.58e-5	2.46e-5
DAX	5	3.76e-5	1.04e-4	2.27e-5	1.18e-4	5.10e-5	1.34e-4
(N=85)	6	3.68e-5	1.02e-4	1.82e-5	1.02e-4	3.22e-5	1.30e-4
	7	3.00e-5	9.85e-5	1.43e-5	9.49e-5	3.26e-5	1.27e-4
	8	2.39e-5	9.41e-5	1.22e-5	9.27e-5	1.94e-5	1.05e-4
	9	2.03e-5	8.62e-5	1.26e-5	8.54e-5	1.86e-5	9.20e-5
	10	2.04e-5	7.66e-5	8.95e-6	8.36e-5	1.22e-5	8.81e-5
FTSE	5	1.22e-4	1.09e-4	6.42e-5	1.58e-4	1.02e-4	1.18e-4
(N=89)	6	8.83e-5	8.75e-5	5.88e-5	1.23e-4	8.45e-5	9.34e-5
	7	6.88e-5	7.46e-5	4.65e-5	9.59e-5	7.13e-5	7.74e-5
	8	6.50e-5	6.37e-5	3.96e-5	9.45e-5	5.31e-5	6.65e-5
	9	4.75e-5	6.17e-5	2.63e-5	8.78e-5	4.22e-5	7.88e-5
	10	4.39e-5	5.96e-5	2.19e-5	7.87e-5	3.49e-5	6.90e-5
S&P	5	8.79e-5	1.06e-4	4.50e-5	1.14e-4	8.40e-5	1.07e-4
(N=98)	6	6.66e-5	8.46e-5	3.63e-5	9.18e-5	8.42e-5	9.62e-5
	7	5.47e-5	7.26e-5	2.99e-5	8.48e-5	5.92e-5	7.58e-5
	8	5.20e-5	5.63e-5	2.74e-5	7.86e-5	5.38e-5	7.55e-5
	9	3.82e-5	5.61e-5	1.95e-5	5.83e-5	4.16e-5	5.74e-5
	10	3.46e-5	4.69e-5	1.94e-5	5.36e-5	3.73e-5	5.19e-5
Nikkei	5	1.18e-4	1.14e-4	7.28e-5	1.27e-4	1.09e-4	1.38e-4
(N=225)	6	7.30e-5	1.01e-4	5.22e-5	1.12e-4	1.06e-4	1.13e-4
	7	7.87e-5	8.19e-5	3.88e-5	9.90e-5	7.25e-5	1.03e-4
	8	6.38e-5	8.43e-5	3.56e-5	9.95e-5	6.29e-5	9.27e-5
	9	4.18e-5	6.17e-5	3.24e-5	9.72e-5	4.79e-5	8.53e-5
	10	4.63e-5	5.93e-5	2.75e-5	9.38e-5	4.62e-5	8.24e-5

to compare our method with the hybrid evolutionary algorithm [29] and the hybrid half thresholding algorithm [2] for index tracking. The computational results demonstrate that our approach generally produces sparse portfolios with smaller out-of-sample tracking error and higher consistency between in-sample and out-of-sample tracking errors. Moreover, our method outperforms the other two approaches in terms of speed.

Acknowledgements This work was supported by the National Natural Science Foundation of China (Grant Nos. 11101325, 11271297, 71371152, 71171158).

References

1 Donoho D L. Compressed sensing. Information Theory, IEEE Transactions on, 2006, 52: 1289-1306

2 Xu F M, Xu Z B, Xue H G. Sparse index tracking: An $L_{1/2}$ regularization based model and solution. Submitted, 2012

3 Tropp J A, Gilbert A C. Signal recovery from random measurements via orthogonal matching pursuit. IEEE Transactions on Information Theory, 2007, 53: 4655-4666

4 Donoho D L, Tsaig Y, Drori I and Starck J C. Sparse solution of underdetermined systems of linear equations by stagewise orthogonal matching pursuit. IEEE Transactions on Information Theory, 2012, 58: 1094-1121

5 Needell D and Tropp J. CoSaMP: Iterative signal recovery from incomplete and inaccurate samples. Applied and Computational Harmonic Analysis, 2009, 26: 301-321

6 Blumensath T and Davies M E. Iterative hard thresholding for compressed sensing. Applied and Computational

Table 4 The comparison on small data sets.

Index	Sparsity (s)	$Cons(APGT\text{-}LS)$	$Cons(MIP)$	$Cons(l_{1/2})$	$SupO(APGT\text{-}LS, MIP)$	$SupO(APGT\text{-}LS, l_{1/2})$
Hang	5	4.10e-6	2.47e-5	7.88e-6	21.2	2.53
Seng	6	**2.95e-5**	**1.32e-5**	**1.21e-5**	27.7	**-4.49**
(N=31)	7	4.69e-6	2.10e-5	8.40e-6	40.0	13.7
	8	5.12e-6	1.91e-5	5.77e-6	29.2	7.41
	9	5.79e-6	1.22e-5	1.02e-5	18.2	12.4
	10	6.11e-6	9.26e-6	8.75e-6	36.7	29.7
DAX	5	6.65e-5	9.57e-5	8.25e-5	12.1	22.0
(N=85)	6	6.52e-5	8.39e-5	9.79e-5	0.14	21.7
	7	6.85e-5	8.06e-5	9.42e-5	**-3.88**	22.3
	8	7.02e-5	8.05e-5	8.58e-5	**-1.47**	10.6
	9	6.60e-5	7.28e-5	7.34e-5	**-1.02**	6.27
	10	5.62e-5	7.46e-5	7.59e-5	8.35	13.1
FTSE	5	1.30e-5	9.40e-5	1.62e-5	31.1	7.41
(N=89)	6	8.37e-7	6.45e-5	8.86e-6	29.0	6.30
	7	5.79e-6	4.93e-5	6.20e-6	22.2	3.57
	8	1.28e-6	5.49e-5	1.35e-5	32.6	4.18
	9	1.41e-5	6.15e-5	3.66e-5	29.8	21.7
	10	1.57e-5	5.68e-5	3.41e-5	24.3	13.7
S&P	5	1.82e-5	6.93e-5	2.28e-5	7.16	0.72
(N=98)	6	**1.80e-5**	5.55e-5	**1.20e-5**	7.84	12.0
	7	**1.78e-5**	5.49e-5	**1.66e-5**	14.5	4.25
	8	4.31e-6	5.12e-5	2.17e-5	28.3	25.4
	9	**1.79e-5**	3.88e-5	**1.58e-5**	3.83	2.29
	10	1.23e-5	3.42e-5	1.46e-5	12.5	9.59
Nikkei	5	3.86e-6	5.42e-5	2.92e-5	10.4	17.8
(N=225)	6	**2.77e-5**	5.99e-5	**6.97e-6**	10.1	11.2
	7	3.14e-6	6.02e-5	3.02e-5	17.3	20.3
	8	2.04e-5	6.40e-5	2.98e-5	15.3	9.11
	9	1.99e-5	6.48e-5	3.74e-5	36.5	27.6
	10	1.29e-5	6.63e-5	3.62e-5	36.8	28.1

Harmonic Analysis, 2000, 27: 265-274
7 Foucart S. Hard thresholding pursuit: an algorithm for compressive sensing. SIAM Journal on Numerical Analysis, 2011, 49: 2543-2563
8 Daubechies I, Mefrise M and Mol C. An iterative thresholding algorithm for linear inverse problems with a sparse constraint. Communication on Pure and Applied Mathematics, 2004, 57: 1413-1457
9 Beck A and Teboule M. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. SIAM Journal on Imaging Sciences, 2009, 2: 183-202
10 Candes E, Romberg J and Tao T. Stable signal recovery from incomplete and inaccurate measurements. Communication on Pure and Applied Mathematics, 2006, 59: 1207-1223
11 Chen S S, Donoho D L and Saunders M A. Atomic decomposition by basis pursuit. SIAM Journal on Scientific Computing, 1998, 20: 33-61
12 Friedman J, Hastie T, Hofling H and Tibshirani R. Pathwise coordinate optimization. The Annals of Applied Statistics, 2007, 1: 302-332
13 Efron B, Hastie T, Johnstone I and Tibshirani R. Least angle regression. The Annals of Statistics, 2004, 32: 407-499
14 Wen Z W, Yin W T, Goladfarb D and Zhang Y. A fast algorithm for sparse reconstruction based on shrinkage, subspace optimization, and continution. SIAM Journal on Scientific Computing, 2010, 32: 1832-1857
15 Wen Z W, Yin W T, Zhang H C and Goldfarb D. On the convergence of an active-set for l_1 minimization. Optimization Methods and Software, 2012, 27: 1127-1146
16 Chen X J, Xu F M and Ye Y Y. Lower bound theory of nonzero entries in the solutions of l_2-l_p minimization. SIAM Journal on Scientific Computing, 2010, 32: 2832-2852
17 Chen X J, Niu L F and Yuan Y X. Optimality conditions and smoothing trust region Newton method for non-Lipschitz

Table 5 The in-sample and out-of-sample tracking errors on large data sets.

Index	Sparsity	APGT-LS		MIP		$l_{1/2}$	
	s	TEI	TEO	TEI	TEO	TEI	TEO
	5	2.74e-5	2.35e-5	1.43e-5	2.77e-5	3.00e-5	2.57e-5
CSI 300	6	2.04e-5	2.34e-5	1.28e-5	2.69e-5	2.48e-5	2.48e-5
(N=300)	7	1.50e-5	1.96e-5	1.17e-5	2.07e-5	2.28e-5	1.94e-5
	8	1.22e-5	1.58e-5	7.85e-6	1.84e-5	1.63e-5	2.29e-5
	9	1.35e-5	1.68e-5	8.03e-6	1.74e-5	1.51e-5	2.15e-5
	10	1.43e-5	1.63e-5	6.74e-6	1.71e-5	1.29e-5	1.74e-5
	20	4.66e-6	7.13e-6	3.45e-6	9.36e-6	6.47e-6	1.04e-5
	30	3.31e-6	6.37e-6	2.48e-6	8.33e-6	6.53e-6	7.08e-6
	40	1.85e-6	4.17e-6	2.74e-6	5.41e-6	5.03e-6	4.77e-6
	50	1.16e-6	3.85e-6	1.88e-6	5.38e-6	5.93e-6	4.06e-6
	10	1.17e-4	2.01e-4	4.67e-5	2.49e-4	1.46e-4	2.78e-4
S&P	20	4.10e-5	1.45e-4	1.89e-5	1.41e-4	6.23e-5	1.55e-4
(N=457)	30	2.81e-5	1.39e-4	1.27e-5	1.67e-4	3.60e-5	1.66e-4
	40	1.70e-5	8.68e-5	6.76e-6	1.22e-4	3.81e-5	9.52e-5
	50	8.35e-6	7.99e-5	5.86e-6	1.38e-4	3.29e-5	1.31e-4
	80	9.58e-7	1.02e-4	3.35e-6	1.07e-4	5.25e-5	1.08e-4
	80	3.20e-6	2.19e-4	6.44e-6	2.31e-4	1.31e-4	2.26e-4
Russell 2000	90	1.42e-6	2.00e-4	7.06e-6	2.10e-4	1.37e-4	2.16e-4
(N=1318)	100	6.94e-7	2.29e-4	6.67e-6	2.33e-4	5.43e-5	2.45e-4
	120	4.38e-8	1.98e-4	3.58e-6	2.39e-4	1.07e-4	2.08e-4
	150	6.41e-7	2.11e-4	1.83e-6	2.27e-4	5.08e-6	3.96e-4
	200	1.82e-6	1.97e-4	1.98e-6	2.70e-4	1.45e-5	3.10e-4
	80	2.15e-6	1.26e-4	7.17e-6	1.36e-4	1.77e-4	1.65e-4
Russell 3000	90	1.14e-6	1.22e-4	6.06e-6	1.53e-4	4.04e-4	1.25e-4
(N=2151)	100	1.00e-6	1.17e-4	4.84e-6	1.30e-4	1.24e-4	1.73e-4
	120	2.72e-7	1.13e-4	3.49e-6	1.27e-4	2.94e-4	1.26e-4
	150	1.98e-6	1.18e-4	2.76e-6	1.47e-4	8.37e-6	1.96e-4
	200	1.02e-6	1.02e-4	3.02e-6	1.29e-4	6.41e-6	2.20e-4

optimization. SIAM Journal on Optimization, 2013, 23: 1528-1552
18 Chen X J, Ng M K and Zhang C. Non-Lipschitz l_p-regularization and box constrained model for image restoration. IEEE Transactions on Image Processing, 2012, 21: 4709-4721
19 Xu Z B Chang X Y, Xu F M and Zhang H. $L_{1/2}$ Regualrization: An iterative half thresholding algorithm. IEEE Transactions on Numerical Networks and Learning Systems, 2012, 23: 1013-1027
20 Kim S, Koh K, Lustig M, Boyd S and Gerinversky D. A method for large-sclae l_1-regularizaed least squares problems with applications in signal processing and statistics. IEEE Journal of Selected Topics in Siganl Processing, 2007
21 Lu Z S and Zhang Y. Sparse approximation via penalty decomposition methods. SIAM Journal on Optimization, 2013
22 Kyrillidis A, Becker S, Cevher Vand and Koch C. Sparse projections onto the simplex. Computer Science of arXiv.org., http://arxiv.org/abs/1206.1529
23 Tibshirani R. Regression shrinkage and selection via the Lasso. Journal of Royal Statistical Society, Series B, 1996, 58: 267-288
24 Candes E J, Romberg J and Tao T. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. IEEE Transactions on Information Theory, 2006, 52: 489-509
25 Blumensath T and Davies M E. Normalized iterative hard thresholding: Guaranteed stability and performance, IEEE Journal of Selected Topics in Signal Processing, 2010, 4: 298-309
26 Xu F M, Lu Z S and Xu Z B. An efficient optimization approach for cardinality-constrained index tracking problems. Submitted, 2013
27 Fan J Q, Zhang J J and Yu K. Asset allocation and risk assessment with gross exposure constraints for vast portfolios. Available at SSRN 1307423, 2008
28 Fastrich B, Paterlini S and Winker P. Cardinality versus q-norm constraints for index tracking. Quantitative Finance, 2012,(ahead-of-print): 1-14
29 Torrubiano P R. and Alberto S. A hybrid optimization approach to index tracking. Annals of Operation Research, 2009, 166: 57-71

Table 6 The comparison on large data sets.

Index	Sparsity	Time			$SupO(APGT\text{-}LS, MIP)$	$SupO(APGT\text{-}LS, l_{1/2})$
	s	APGT-LS	MIP	$l_{1/2}$		
	5	0.133	4.38	1.92	15.4	8.66
CSI 300	6	0.136	4.74	1.99	12.8	5.48
(N=300)	7	0.134	4.75	1.93	5.57	**-0.74**
	8	0.137	4.74	2.40	13.7	30.8
	9	0.137	4.83	1.95	3.20	21.8
	10	0.143	4.85	1.93	4.73	6.25
	20	0.157	2.40	1.96	23.8	31.1
	30	0.204	2.04	1.94	23.5	9.97
	40	0.255	1.92	1.98	22.9	12.7
	50	0.388	1.38	1.97	28.4	5.16
	10	0.080	5.70	1.60	19.1	27.8
S&P	20	0.094	7.10	1.62	**-2.99**	6.25
(N=457)	30	0.123	10.35	1.62	17.0	16.1
	40	0.160	13.88	1.66	28.9	27.1
	50	0.245	22.12	1.61	**-0.18**	**-0.19**
	80	0.739	34.29	1.66	4.36	5.22
	80	0.725	63.04	18.3	5.07	3.13
Russell 2000	90	1.014	87.70	18.9	4.63	7.50
(N=1318)	100	1.197	117.63	19.1	1.97	6.70
	120	2.460	201.79	18.9	17.2	4.76
	150	2.270	472.37	19.8	7.12	46.7
	200	2.063	1258.45	22.9	26.9	36.3
	80	0.981	86.77	29.8	7.58	23.6
Russell 3000	90	1.124	113.17	29.9	20.1	2.06
(N=2151)	100	1.459	147.19	30.1	9.70	32.4
	120	3.48	256.56	29.6	10.9	10.4
	150	2.41	572.74	36.2	20.0	39.9
	200	2.95	1562.13	41.1	20.3	53.4

30 Beasley J E. OR-Library: Distributing test problems by electronic mail. Journal of the Operational Research Society, 1990, 41: 1069-1072