Math. Program., Ser. B (2011) 127:123–144
DOI 10.1007/s10107-010-0416-0

FULL LENGTH PAPER

Testing the nullspace property using semidefinite programming

Alexandre d'Aspremont · Laurent El Ghaoui

Received: 16 February 2009 / Accepted: 26 November 2009 / Published online: 12 October 2010

Abstract Recent results in compressed sensing show that, under certain conditions, the sparsest solution to an underdetermined set of linear equations can be recovered by solving a linear program. These results either rely on computing sparse eigenvalues of the design matrix or on properties of its nullspace. So far, no tractable algorithm is known to test these conditions and most current results rely on asymptotic properties of random matrices. Given a matrix A, we use semidefinite relaxation techniques to test the nullspace property on A and show on some numerical examples that these relaxation bounds can prove perfect recovery of sparse solutions with relatively high cardinality.

Keywords Compressed sensing · Nullspace property · Semidefinite programming · Restricted isometry constant

Mathematics Subject Classification (2000) 90C22 · 94A12 · 90C27

1 Introduction

A recent stream of results in signal processing have focused on producing explicit conditions under which the sparsest solution to an underdetermined linear system can be found by solving a linear program. Given a matrix $A \in \mathbf{R}^{m \times n}$ with $n > m$ and a vector $v \in \mathbf{R}^m$, writing $\|x\|_0 = \mathbf{Card}(x)$ the number of nonzero coefficients in x, this

A. d'Aspremont (✉)
ORFE Department, Princeton University, Princeton, NJ 08544, USA
e-mail: aspremon@princeton.edu

L. El Ghaoui
EECS Department, U.C. Berkeley, Berkeley, CA 94720, USA
e-mail: elghaoui@eecs.berkeley.edu

means that the solution of the following (combinatorial) ℓ_0 minimization problem:

$$\begin{array}{ll}\text{minimize} & \|x\|_0\\ \text{subject to} & Ax = v,\end{array} \tag{1}$$

in the variable $x \in \mathbf{R}^n$, can be found by solving the (convex) ℓ_1 minimization problem:

$$\begin{array}{ll}\text{minimize} & \|x\|_1\\ \text{subject to} & Ax = v,\end{array} \tag{2}$$

in the variable $x \in \mathbf{R}^n$, which is equivalent to a linear program.

Based on results by [23] and [1,11] show that when the solution x_0 of (1) is sparse with $\mathbf{Card}(x_0) = k$ and the coefficients of A are i.i.d. Gaussian, then the solution of the ℓ_1 problem in (2) will always match that of the ℓ_0 problem in (1) provided k is below an explicitly computable *strong recovery* threshold k_S. They also show that if k is below another (larger) *weak recovery* threshold k_W, then these solutions match with an exponentially small probability of failure.

Universal conditions for strong recovery based on sparse extremal eigenvalues were derived in [5] and [4] who also proved that certain (mostly random) matrix classes satisfied these conditions with an exponentially small probability of failure. Simpler, weaker conditions which can be traced back to [9,24] or [6] for example, are based on properties of the nullspace of A. In particular, if we define

$$\alpha_k = \max_{\{Ax=0,\ \|x\|_1=1\}} \ \max_{\{\|y\|_\infty=1,\ \|y\|_1\le k\}} y^T x,$$

these references show that $\alpha_k < 1/2$ guarantees strong recovery.

One key issue with the current sparse recovery conditions in [5] or [9] is that except for explicit recovery thresholds available for certain types of random matrices, testing these conditions on generic matrices is potentially *harder* than solving the combinatorial ℓ_0-norm minimization problem in (1) for example as it implies either solving a combinatorial problem to compute α_k, or computing sparse eigenvalues. Semidefinite relaxation bounds on sparse eigenvalues were used in [8] or [15] for example to test the restricted isometry conditions in [5] on arbitrary matrices. In recent independent results, [14] provide an alternative proof of some of the results in [9], extend them to the noisy case and produce a linear programming (LP) relaxation bound on α_k with explicit performance bounds.

In this paper, we derive a semidefinite relaxation bound on α_k, study its tightness and performance. By randomization, the semidefinite relaxation also produces lower bounds on the objective value as a natural by-product of the solution. Overall, our bounds are slightly better than LP ones numerically but both relaxations share the same asymptotic performance limits. However, because it involves solving a semidefinite program, the complexity of the semidefinite relaxation derived here is significantly higher than that of the LP relaxation.

The paper is organized as follows. In Sect. 2, we briefly recall some key results in [9] and [6]. We derive a semidefinite relaxation bound on α_k in Sect. 3, and study its tightness and performance in Sect. 4. Section 5 describes a first-order algorithm to solve the resulting semidefinite program. Finally, we test the numerical performance of this relaxation in Sect. 6.

Notation To simplify notation here, for a matrix $X \in \mathbf{R}^{m\times n}$, we write its columns X_i, $\|X\|_1$ the sum of absolute values of its coefficients (not the ℓ_1 norm of its spectrum) and $\|X\|_\infty$ the largest coefficient magnitude. More classically, $\|X\|_F$ and $\|X\|_2$ are the Frobenius and spectral norms.

2 Sparse recovery and the null space property

Given a *coding* matrix $A \in \mathbf{R}^{m\times n}$ with $n > m$, a *sparse* signal $x_0 \in \mathbf{R}^n$ and an information vector $v \in \mathbf{R}^m$ such that

$$v = Ax_0,$$

we focus on the problem of perfectly recovering the signal x_0 from the vector v, assuming the signal x_0 is sparse enough. We define the decoder $\Delta_1(v)$ as a mapping from $\mathbf{R}^m \rightarrow \mathbf{R}^n$, with

$$\Delta_1(v) \triangleq \underset{\{x\in\mathbf{R}^n:\ Ax=v\}}{\operatorname{argmin}} \|x\|_1. \tag{3}$$

This particular decoder is equivalent to a linear program which can be solved efficiently. Suppose that the original signal x_0 is sparse, a natural question to ask is then: When does this decoder perfectly recover a sparse signal x_0? Recent results by [5], [11] and [6] provide a somewhat tight answer. In particular, as in [6], for a given coding matrix $A \in \mathbf{R}^{m\times n}$ and $k > 0$, we can quantify the ℓ_1 error of a decoder $\Delta(v)$ by computing the smallest constant $C > 0$ such that

$$\|x - \Delta(Ax)\|_1 \leq C\sigma_k(x) \tag{4}$$

for all $x \in \mathbf{R}^n$, where

$$\sigma_k(x) \triangleq \min_{\{z\in\mathbf{R}^n:\ \mathbf{Card}(z)=k\}} \|x - z\|_1$$

is the ℓ_1 error of the best k-term approximation of the signal x and can simply be computed as the ℓ_1 norm of the $n - k$ smallest coefficients of $x \in \mathbf{R}^n$. We now define the *nullspace property* as in [9] or [6].

Definition 1 A matrix $A \in \mathbf{R}^{m\times n}$ satisfies the null space property in ℓ_1 of order k with constant C_k if and only if

$$\|z\|_1 \leq C_k \|z_{T^c}\|_1 \tag{5}$$

holds for all $z \in \mathbf{R}^n$ with $Az = 0$ and index subsets $T \subset [1, n]$ of cardinality $\mathbf{Card}(T) \leq k$, where T^c is the complement of T in $[1, n]$.

[6] for example show the following theorem linking the optimal decoding quality on sparse signals and the nullspace property constant C_k.

Theorem 1 *Given a coding matrix $A \in \mathbf{R}^{m \times n}$ and a sparsity target $k > 0$. If A has the nullspace property in* (5) *of order $2k$ with constant $C/2$, then there exists a decoder Δ_0 which satisfies* (4) *with constant C. Conversely, if* (4) *holds with constant C then A has the nullspace property at the order $2k$ with constant C.*

Proof See [6, Corollary 3.3].

This last result means that the existence of an optimal decoder staisfying (4) is equivalent to A satisfying (5). Unfortunately, this optimal decoder $\Delta_0(v)$ is defined as

$$\Delta_0(v) \triangleq \underset{\{z \in \mathbf{R}^n :\, Az = v\}}{\operatorname{argmin}} \quad \sigma_k(z)$$

hence requires solving a combinatorial problem which is potentially intractable. However, using tighter restrictions on the nullspace property constant C_k, we get the following result about the linear programming decoder $\Delta_1(v)$ in (3).

Theorem 2 *Given a coding matrix $A \in \mathbf{R}^{m \times n}$ and a sparsity target $k > 0$. If A has the nullspace property in* (5) *of order k with constant $C < 2$, then the linear programming decoder $\Delta_1(y)$ in* (3) *satisfies the error bounds in* (4) *with constant $2C/(2 - C)$ at the order k.*

Proof See steps (4.3) to (4.10) in the proof of [6, Theorem 4.3].

To summarize the results above, if there exists a $C > 0$ such that the coding matrix A satisfies the nullspace property in (5) at the order k then there exists a decoder which perfectly recovers signals x_0 with cardinality $k/2$. If, in addition, we can show that $C < 2$, then the linear programming based decoder in (3) perfectly recovers signals x_0 with cardinality k. In the next section, we produce upper bounds on the constant C_k in (5) using semidefinite relaxation techniques.

3 Semidefinite relaxation

Given $A \in \mathbf{R}^{m \times n}$ and $k > 0$, we look for a constant $C_k \geq 1$ in (5) such that

$$\|x_T\|_1 \leq (C_k - 1)\|x_{T^c}\|_1$$

for all vectors $x \in \mathbf{R}^n$ with $Ax = 0$ and index subsets $T \subset [1, n]$ with cardinality k. We can rewrite this inequality

$$\|x_T\|_1 \leq \alpha_k \|x\|_1 \tag{6}$$

with $\alpha_k \in [0, 1)$. Because $\alpha_k = 1 - 1/C_k$, if we can show that $\alpha_k < 1$ then we prove that A satisfies the nullspace property at order k with constant C_k. Furthermore, if we prove $\alpha_k < 1/2$, we prove the existence of a linear programming based decoder which perfectly recovers signals x_0 with at most k errors. By homogeneity, the constant α_k can be computed as

$$\alpha_k = \max_{\{Ax=0,\ \|x\|_1=1\}} \ \max_{\{\|y\|_\infty=1,\ \|y\|_1\leq k\}} y^T x, \tag{7}$$

where the equality $\|x\|_1 = 1$ can, without loss of generality, be replaced by $\|x\|_1 \leq 1$. We now derive a semidefinite relaxation for problem (7) as follows. After a change of variables

$$\begin{pmatrix} X & Z^T \\ Z & Y \end{pmatrix} = \begin{pmatrix} xx^T & xy^T \\ yx^T & yy^T \end{pmatrix},$$

we can rewrite (7) as

$$\begin{array}{ll} \text{maximize} & \mathbf{Tr}(Z) \\ \text{subject to} & AXA^T = 0,\ \|X\|_1 \leq 1, \\ & \|Y\|_\infty \leq 1,\ \|Y\|_1 \leq k^2,\ \|Z\|_1 \leq k, \\ & \begin{pmatrix} X & Z^T \\ Z & Y \end{pmatrix} \succeq 0,\ \ \mathbf{Rank}\begin{pmatrix} X & Z^T \\ Z & Y \end{pmatrix} = 1, \end{array} \tag{8}$$

in the variables $X, Y \in \mathbf{S}_n$, $Z \in \mathbf{R}^{n\times n}$, where all norms should be understood componentwise. We then simply drop the rank constraint to form a relaxation of (7) as

$$\begin{array}{ll} \text{maximize} & \mathbf{Tr}(Z) \\ \text{subject to} & AXA^T = 0,\ \|X\|_1 \leq 1, \\ & \|Y\|_\infty \leq 1,\ \|Y\|_1 \leq k^2,\ \|Z\|_1 \leq k, \\ & \begin{pmatrix} X & Z^T \\ Z & Y \end{pmatrix} \succeq 0, \end{array} \tag{9}$$

which is a semidefinite program in the variables $X, Y \in \mathbf{S}_n$, $Z \in \mathbf{R}^{n\times n}$. Note that the contraint $\|Z\|_1 \leq k$ is redundant in the rank one problem but not in its relaxation. Because all constraints are linear here, dropping the rank constraint is equivalent to computing a Lagrangian (bidual) relaxation of the original problem and adding redundant constraints to the original problem often tightens these relaxations. The dual of program (9) can be written

$$\begin{array}{ll} \text{minimize} & \|U_1\|_\infty + k^2\|U_2\|_\infty + \|U_3\|_1 + k\|U_4\|_\infty \\ \text{subject to} & \begin{pmatrix} U_1 - A^T WA & -\frac{1}{2}(\mathbf{I} + U_4) \\ -\frac{1}{2}(\mathbf{I} + U_4^T) & U_2 + U_3 \end{pmatrix} \succeq 0, \end{array}$$

which is a semidefinite program in the variables $U_1, U_2, U_3, W \in \mathbf{S}_n$ and $U_4 \in \mathbf{R}^{n\times n}$. For any feasible point of this program, the objective $\|U_1\|_\infty + k^2\|U_2\|_\infty + \|U_3\|_1 + k\|U_4\|_\infty$ is an upper bound on the optimal value of (9), hence on α_k. We can further simplify this program using elimination results for LMIs. In fact, [3, §2.6.2] shows that this last problem is equivalent to

$$\begin{array}{ll}\text{minimize} & \|U_1\|_\infty + k^2\|U_2\|_\infty + \|U_3\|_1 + k\|U_4\|_\infty \\ \text{subject to} & \begin{pmatrix} U_1 - wA^TA & -\frac{1}{2}(\mathbf{I}+U_4) \\ -\frac{1}{2}(\mathbf{I}+U_4^T) & U_2+U_3 \end{pmatrix} \succeq 0, \end{array} \tag{10}$$

where the variable w is now scalar. In fact, using the same argument, letting $P \in \mathbf{R}^{n\times p}$ be an orthogonal basis of the nullspace of A, i.e. such that $AP = 0$ with $P^TP = \mathbf{I}$, we can rewrite the previous problem as follows

$$\begin{array}{ll}\text{minimize} & \|U_1\|_\infty + k^2\|U_2\|_\infty + \|U_3\|_1 + k\|U_4\|_\infty \\ \text{subject to} & \begin{pmatrix} P^TU_1P & -\frac{1}{2}P^T(\mathbf{I}+U_4) \\ -\frac{1}{2}(\mathbf{I}+U_4^T)P & U_2+U_3 \end{pmatrix} \succeq 0, \end{array} \tag{11}$$

which is a (smaller) semidefinite program in the variables $U_1, U_2, U_3 \in \mathbf{S}_n$ and $U_4 \in \mathbf{R}^{n\times n}$. The dual of this last problem is then

$$\begin{array}{ll}\text{maximize} & \mathbf{Tr}(Q_2^TP) \\ \text{subject to} & \|PQ_1P^T\|_1 \le 1,\ \|PQ_2^T\|_1 \le k \\ & \|Q_3\|_\infty \le 1,\ \|Q_3\|_1 \le k^2 \\ & \begin{pmatrix} Q_1 & Q_2^T \\ Q_2 & Q_3 \end{pmatrix} \succeq 0, \end{array} \tag{12}$$

which is a semidefinite program in the matrix variables $Q_1 \in \mathbf{S}_p,\ Q_2 \in \mathbf{R}^{p\times n},\ Q_3 \in \mathbf{S}_n$, whose objective value is equal to that of problem (9).

Note that adding any number of redundant constraints in the original problem (8) will further improve tightness of the semidefinite relaxation, at the cost of increased complexity. In particular, we can use the fact that when

$$\|x\|_1 = 1,\ \|y\|_\infty = 1,\ \|y\|_1 \le k,$$

and if we set $Y = yy^T$ and $Z = yx^T$, we must have

$$\sum_{i=1}^n |Y_{ij}| \le kt_j,\ |Y_{ij}| \le t_j,\ \mathbf{1}^Tt \le k,\ t \le \mathbf{1},\quad \text{for } i, j = 1, \dots, n,$$

and

$$\sum_{i=1}^{n} |Z_{ij}| \le k r_j, \ |Z_{ij}| \le r_j, \ \mathbf{1}^T r \le k, \quad \text{for } i, j = 1, \dots, n,$$

for $r, t \in \mathbf{R}^n$. This means that we can refine the constraint $\|Z\|_1 \le k$ in (9) to solve instead

$$\begin{array}{ll} \text{maximize} & \mathbf{Tr}(Z) \\ \text{subject to} & AXA^T = 0, \ \|X\|_1 \le 1, \\ & \sum_{i=1}^{n} |Y_{ij}| \le k t_j, \quad |Y_{ij}| \le t_j, \ \mathbf{1}^T t \le k, \quad t \le \mathbf{1}, \\ & \sum_{i=1}^{n} |Z_{ij}| \le k r_j, \quad |Z_{ij}| \le r_j, \ \mathbf{1}^T r \le 1, \quad \text{for } i, j = 1, \dots, n, \\ & \begin{pmatrix} X & Z^T \\ Z & Y \end{pmatrix} \succeq 0, \end{array} \tag{13}$$

which is a semidefinite program in the variables $X, Y \in \mathbf{S}_n$, $Z \in \mathbf{R}^{n \times n}$ and $r, t \in \mathbf{R}^n$. Adding these columnwise constraints on Y and Z significantly tightens the relaxation. Any *feasible* solution to the dual of (13) with objective value less than $1/2$ will then be a certificate that $\alpha_k < 1/2$.

4 Tightness and limits of performance

The relaxation above naturally produces a covariance matrix as its output and we use randomization techniques as in [12] to produce primal solutions for problem (7). Then, following results by A. Nemirovski (private communication), we bound the performance of the relaxation in (9).

4.1 Randomization

Here, we show that lower bounds on α_k can be generated as a natural by-product of the relaxation. We use solutions to the semidefinite program in (9) and generate feasible points to (7) by randomization. These can then be used to certify that $\alpha_k > 1/2$ and prove that a matrix does not satisfy the nullspace property. Suppose that the matrix

$$\Gamma = \begin{pmatrix} X & Z^T \\ Z & Y \end{pmatrix} \tag{14}$$

solves problem (9), because $\Gamma \succeq 0$, we can generate Gaussian variables $(x, y) \sim \mathcal{N}(0, \Gamma)$. Below, we show that after proper scaling, (x, y) will satisfy the constraints of problem (7) with high probability, and use this result to quantify the quality of these randomized solutions. We begin by recalling classical results on the moments of $\|x\|_1$ and $\|x\|_\infty$ when $x \sim \mathcal{N}(0, X)$ and bound deviations above their means using concentration inequalities on Lipschitz functions of Gaussian variables.

Lemma 1 *Let* $X \in \mathbf{S}_n$, $x \sim \mathcal{N}(0, X)$ *and* $\delta > 0$, *we have*

$$\mathbf{P}\left(\frac{\|x\|_1}{(\sqrt{2/\pi} + \sqrt{2\log\delta})\sum_{i=1}^n (X_{ii})^{1/2}} \geq 1\right) \leq \frac{1}{\delta} \tag{15}$$

Proof Let P be the square root of X and $u_i \sim \mathcal{N}(0,1)$ be independent Gaussian variables, we have

$$\|x\|_1 = \sum_{i=1}^n \left|\sum_{j=1}^n P_{ij}u_j\right|$$

hence, because each term $|\sum_{j=1}^n P_{ij}u_j|$ is a Lipschitz continuous function of the variables u with constant $(\sum_{j=1}^n P_{ij}^2)^{1/2} = (X_{ii})^{1/2}$, $\|x\|_1$ is Lipschitz with constant $L = \sum_{i=1}^n (X_{ii})^{1/2}$. Using the concentration inequality by [13] (see also [16] for a general discussion) we get for any $\beta > 0$

$$\mathbf{P}\left(\frac{\|x\|_1}{\beta} \geq \frac{\mathbf{E}[\|x\|_1] + t}{\beta}\right) \leq \exp\left(-\frac{t^2}{2L^2}\right)$$

with $\mathbf{E}[\|x\|_1] = \sqrt{2/\pi}\sum_{i=1}^n (X_{ii})^{1/2}$. Picking $t = \sqrt{2\log\delta}L$ and $\beta = \mathbf{E}[\|x\|_1] + t$ yields the desired result.

We now recall another classic result on the concentration of $\|y\|_\infty$, also based on the fact that $\|y\|_\infty$ is a Lipschitz continuous function of independent Gaussian variables.

Lemma 2 *Let* $Y \in \mathbf{S}_n$, $y \sim \mathcal{N}(0, Y)$ *and* $\delta > 0$ *then*

$$\mathbf{P}\left(\frac{\|y\|_\infty}{(\sqrt{2\log 2n} + \sqrt{2\log\delta})\max_{i=1,\ldots,n}(Y_{ii})^{1/2}} \geq 1\right) \leq \frac{1}{\delta} \tag{16}$$

Proof [16, Theorem 3.12] shows that $\|y\|_\infty$ is a Lipschitz function of independent Gaussian random variables with constant $\max_{i=1,\ldots,n}(Y_{ii})^{1/2}$, hence a reasoning similar to that in lemma 1 yields the desired result.

Using union bounds, the lemmas above show that if we pick $3/\delta < 1$ and $(x, y) \sim \mathcal{N}(0, \Gamma)$, the scaled sample points

$$\left(\frac{x}{g(X,\delta)}, \frac{y}{h(Y,n,k,\delta)}\right)$$

will be feasible in (7) with probability at least $1 - 3/\delta$ if we set

$$g(X,\delta) = \left(\sqrt{2/\pi} + \sqrt{2\log\delta}\right)\sum_{i=1}^n (X_{ii})^{1/2} \tag{17}$$

and

$$\begin{aligned} &h(Y, n, k, \delta) \\ &= \max \left\{ \left(\sqrt{2 \log 2n} + \sqrt{2 \log \delta}\right) \max_{i=1,\ldots,n} (Y_{ii})^{1/2}, \right. \\ &\quad \left. \frac{\left(\sqrt{2/\pi} + \sqrt{2 \log \delta}\right) \sum_{i=1}^{n} (Y_{ii})^{1/2}}{k} \right\} \end{aligned} \tag{18}$$

The randomization technique is then guaranteed to produce a feasible point of (7) with objective value

$$\frac{q_{\{1-3/\delta\}}}{g(X, \delta) h(Y, n, k, \delta)}$$

where $q_{\{1-3/\delta\}}$ is the $1 - 3/\delta$ quantile of $x^T y$ when $(x, y) \sim \mathcal{N}(0, \Gamma)$. We now compute a (relatively coarse) lower bound on the value of that quantile.

Lemma 3 *Let* $\epsilon,\ \delta > 3$ *and* $(x, y) \sim \mathcal{N}(0, \Gamma)$*, with* Γ *defined as in (14), then*

$$\mathbf{P}\left(\sum_{i=1}^{n} x_i y_i \geq \mathbf{Tr}(Z) - \frac{\sqrt{3}}{\sqrt{\delta - 3}} \sigma\right) \geq \frac{3}{\delta} \tag{19}$$

where

$$\sigma^2 = \|Z\|_F^2 + \mathbf{Tr}(XY).$$

Proof Let $S \in \mathbf{R}^{2n \times 2n}$ be such that $\Gamma = S^T S$ and $(x, y) \sim \mathcal{N}(0, \Gamma)$, we have

$$\mathbf{E}\left[\left(y^T \quad x\right)^2\right] = \sum_{i,j=1}^{n} \mathbf{E}\left[\left(S_i^T w\right)\left(S_{n+i}^T w\right)\left(S_j^T w\right)\left(S_{n+j}^T w\right)\right]$$

where w is a standard normal vector of dimension $2n$. Wick's formula implies

$$\begin{aligned} \mathbf{E}\left[(S_i^T w)(S_{n+i}^T w)(S_j^T w)(S_{n+j}^T w)\right] &= \mathbf{Haf}\begin{pmatrix} X_{ii} & Z_{ii} & X_{ij} & Z_{ij} \\ Z_{ii} & Y_{ii} & Z_{ij} & Y_{ij} \\ X_{ij} & Z_{ij} & X_{jj} & Z_{jj} \\ Z_{ij} & Y_{ij} & Z_{jj} & Y_{jj} \end{pmatrix} \\ &= Z_{ii} Z_{jj} + Z_{ij}^2 + X_{ij} Y_{ij}, \end{aligned}$$

where $\mathbf{Haf}(X)$ is the Hafnian of the matrix X (see [2] for example), which means

$$\mathbf{E}\left[(y^T x)^2\right] = (\mathbf{Tr}(Z))^2 + \|Z\|_F^2 + \mathbf{Tr}(XY).$$

Because $\mathbf{E}[y^T x] = \mathbf{E}[\mathbf{Tr}(xy^T)] = \mathbf{Tr}(\mathbf{E}[xy^T]) = \mathbf{Tr}(Z)$, we then conclude using Cantelli's inequality, which gives

$$\mathbf{P}\left(\sum_{i=1}^{n} x_i y_i \leq \mathbf{Tr}(Z) - t\sigma\right) \leq \frac{1}{1+t^2}$$

having set $t = \sqrt{3}/\sqrt{\delta - 3}$.

We can now combine these results to produce a lower bound on the objective value achieved by randomization.

Theorem 3 *Given $A \in \mathbf{R}^{m\times n}$, $\epsilon > 0$ and $k > 0$, writing SDP_k the optimal value of (9), we have*

$$\frac{SDP_k - \epsilon}{g(X,\delta)h(Y,n,k,\delta)} \leq \alpha_k \leq SDP_k \tag{20}$$

where

$$\delta = 3 + \frac{3\left(\|Z\|_F^2 + \mathbf{Tr}\,(XY)\right)}{\epsilon^2}.$$

$$g(X,\delta) = \left(\sqrt{2/\pi} + \sqrt{2\log\delta}\right)\sum_{i=1}^{n}(X_{ii})^{1/2}$$

and

$$\begin{aligned} &h(Y,n,k,\delta)\\ &= \max\left\{\left(\sqrt{2\log 2n} + \sqrt{2\log\delta}\right)\max_{i=1,\ldots,n}(Y_{ii})^{1/2},\right.\\ &\qquad\left.\frac{\left(\sqrt{2/\pi}+\sqrt{2\log\delta}\right)\sum_{i=1}^{n}(Y_{ii})^{1/2}}{k}\right\}\end{aligned}$$

Proof If Γ solves (9) and the vectors (x, y) are sampled according to $(x, y) \sim \mathcal{N}(0,\Gamma)$, then

$$\mathbf{E}[(Ax)(Ax)^T] = \mathbf{E}[Axx^TA^T] = AXA^T = 0,$$

means that we always have $Ax = 0$. When $\delta > 3$, Lemmas 1 and 2 show that

$$\left(\frac{x}{g(X,\delta)}, \frac{y}{h(Y,n,k,\delta)}\right)$$

will be feasible in (7) with probability at least $1 - 3/\delta$, hence we can get a feasible point for (7) by sampling enough variables (x, y). Lemma 3 shows that if we set δ as above, the randomization procedure is guaranteed to reach an objective value $y^T x$ at least equal to

$$\frac{\mathbf{Tr}(Z) - \epsilon}{g(X, \delta)h(Y, n, k, \delta)}$$

which is the desired result.

Note that because $\Gamma \succeq 0$, we have $Z_{ij}^2 \leq X_{ii}Y_{jj}$, hence $\|Z\|_F^2 \leq \mathbf{Tr}(X)\,\mathbf{Tr}(Y) \leq k^2$. We also have $\mathbf{Tr}(XY) \leq \|X\|_1 \|Y\|_1 \leq k^2$ hence

$$\delta \leq 3 + \frac{6k^2}{\epsilon^2}.$$

and the only a priori unknown terms controlling tightness are $\sum_{i=1}^n (X_{ii})^{1/2}$, $\sum_{i=1}^n (Y_{ii})^{1/2}$ and $\max_{i=1,\ldots,n}(Y_{ii})^{1/2}$. Unfortunately, while the third term is bounded by one, the first two can become quite large, with trivial bounds giving

$$\sum_{i=1}^n (X_{ii})^{1/2} \leq \sqrt{n} \quad \text{and} \quad \sum_{i=1}^n (Y_{ii})^{1/2} \leq \sqrt{n},$$

which means that, in the worst case, our lower bound will be off by a factor $1/n$. However, we will observe in Sect. 6 that, when $k = 1$, these terms are sometimes much lower than what the worst-case bounds seem to indicate. The expression for the tightness coefficient γ in (14) also highlights the importance of the constraint $\|Z\|_1 \leq k$. Indeed, the positive semidefinitess of 2×2 principal submatrices means that $Z_{ij}^2 \leq X_{ii}Y_{jj}$, hence

$$\|Z\|_1 \leq \left(\sum_{i=1}^n (X_{ii})^{1/2}\right)\left(\sum_{i=1}^n (Y_{ii})^{1/2}\right),$$

so controlling $\|Z\|_1$ potentially tightens the relaxation. This is confirmed in numerical experiments: the relaxation including the (initially) redundant norm constraint on Z is significantly tighter on most examples. Finally, note that better lower bounds on α_k can be obtained (numerically) by sampling $\|x_T\|_1/\|x\|_1$ in (6) directly, or as suggested by one of the referees, solving

$$\begin{array}{ll} \text{maximize} & c^T x \\ \text{subject to} & Ax = 0, \ \|x\|_1 \leq 1, \end{array}$$

in $x \in \mathbf{R}^n$ for various random vectors $c \in \{-1, 0, 1\}^n$ with at most k nonzero coefficients. In both cases unfortunately, the moments cannot be computed explicitly so studying performance is much harder.

4.2 Performance

Following results by A. Nemirovski (private communication), we can derive precise bounds on the performance of the relaxation in (9).

Lemma 4 *Suppose* (X, Y, Z) *solve the semidefinite program in (9), then*

$$\mathbf{Tr}(Z) = \alpha_1$$

and the relaxation is tight for $k = 1$.

Proof First, notice that when the matrices (X, Y, Z) solve (9), $AX = 0$ with

$$\begin{pmatrix} X & Z^T \\ Z & Y \end{pmatrix} \succeq 0$$

means that the rows of Z also belong to the nullspace of A. If A satisfies the nullspace property in (6), we must have $|Z_{ii}| \leq \alpha_1 \sum_{j=1}^n |Z_{ij}|$ for $i = 1, \ldots, n$, hence $\mathbf{Tr}(Z) \leq \alpha_1 \|Z\|_1 \leq \alpha_1$. By construction, we always have $\mathbf{Tr}(Z) \geq \alpha_1$ hence $\mathbf{Tr}(Z) = \alpha_1$ when Z solves (9) with $k = 1$.

As in [14], this also means that if a matrix A satisfies the restricted isometry property at cardinality $O(m)$ (as Gaussian matrices do for example), then the relaxation in (9) will certify $\alpha_k < 1/2$ for $k = O(\sqrt{m})$. Unfortunately, the results that follow show that this is the best we can hope for here.

Without loss of generality, we can assume that $n = 2m$ (if $n \geq 2m$, the problem is harder). Let Q be an orthoprojector on a $(n - m)$-dimensional subspace of the nullspace of A, with $\mathbf{Rank}(Q) = n - m = m$. By construction, $\|Q\|_1 \leq n\|Q\|_2 = n\sqrt{n}$, $0 \preceq Q \preceq \mathbf{I}$ and of course $AQ = 0$. We can use this matrix to construct a feasible solution to problem (13) when $k = \sqrt{m}$. We set $X = Q/\sqrt{n}$, $Y = Q/(n\sqrt{m})$, $Z = Q/n$, $t_j = 1/(n\sqrt{m})$ and $r_j = 1/n$ for $j = 1, \ldots, n$. We then have

$$\|Y_i\|_1 = \frac{\|Q_i\|_1}{n\sqrt{m}} \leq \frac{\|Q_i\|_2}{\sqrt{nm}} \leq \frac{1}{\sqrt{nm}} \leq kt_i, \quad i = 1, \ldots, n,$$

and $\|Y_i\|_\infty \leq \|Y_i\|_2 \leq 1/(n\sqrt{m})$ with $\mathbf{1}^T t \leq k$. We also get

$$\|Z_i\|_1 = \frac{\|Q_i\|_1}{n} \leq \frac{\|Q_i\|_2}{\sqrt{n}} \leq kr_i, \quad i = 1, \ldots, n.$$

With

$$\begin{pmatrix} n^{-1/2} & n^{-1} \\ n^{-1} & n^{-1}m^{-1/2} \end{pmatrix} \succeq 0,$$

the matrices we have defined above form a *feasible point* of problem (13). Because, $\mathbf{Tr}(Z) = \mathbf{Tr}(Q)/n = 1/2$, this feasible point proves that the optimal value of (13) is larger than $1/2$ when $n = 2m$ and $k = \sqrt{m}$. This means that the relaxation in (13) can prove that a matrix satisfies the nullspace property for cardinalities at most $k = O(\sqrt{m})$ and this performance bound is tight since we have shown that it achieves this rate of $O(\sqrt{m})$ for good matrices.

This counter example also produces bounds on the performance of another relaxation for testing sparse recovery. In fact, if we set $X = Q/m$ with Q defined as above, we have $\mathbf{Tr}(X) = 1$ with $X \succeq 0$ and

$$\|X\|_1 = \frac{\|Q\|_1}{m} \leq 2\sqrt{m}$$

and X is an optimal solution of the problem

$$\begin{array}{ll} \text{minimize} & \mathbf{Tr}(XAA^T) \\ \text{subject to} & \|X\|_1 \leq 2\sqrt{2m} \\ & \mathbf{Tr}(X) = 1, \;\; X \succeq 0, \end{array}$$

which is a semidefinite relaxation used in [7,8] to bound the restricted isometry constant $\delta_k(A)$. Because $\mathbf{Tr}(XAA^T) = 0$ by construction, we know that this last relaxation will fail to show $\delta_k(A) < 1$ whenever $k = O(\sqrt{m})$. Somewhat strikingly, this means that the three different tractable tests for sparse recovery conditions, derived in [8,14] and this paper, are *all* limited to showing recovery at the (suboptimal) rate $k = O(\sqrt{m})$.

5 Algorithms

Small instances of the semidefinite program in (11) and be solved efficiently using solvers such as SEDUMI [21] or SDPT3 [22]. For larger instances, it is more advantageous to solve (11) using first order techniques, given a fixed target for α. We set $P \in \mathbf{R}^{n \times p}$ to be an orthogonal basis of the nullspace of the matrix A in (6), i.e. such that $AP = 0$ with $P^T P = \mathbf{I}$. We also let $\bar{\alpha}$ be a target critical value for α (such as $1/2$ for example), and solve the following problem

$$\begin{array}{ll} \text{maximize} \; \lambda_{\min} \begin{pmatrix} P^T U_1 P & -\frac{1}{2} P^T (\mathbf{I} + U_4) \\ -\frac{1}{2}(\mathbf{I} + U_4^T) P & U_2 + U_3 \end{pmatrix} \\ \text{subject to} \;\; \|U_1\|_\infty + k^2 \|U_2\|_\infty + \|U_3\|_1 + k\|U_4\|_\infty \leq \bar{\alpha} \end{array} \tag{21}$$

in the variables $U_1, U_2, U_3 \in \mathbf{S}_n$ and $U_4 \in \mathbf{R}^{n \times n}$. If the objective value of this last problem is greater than zero, then the optimal value of problem (11) is necessarily smaller than $\bar{\alpha}$, hence $\alpha \leq \bar{\alpha}$ in (7).

Because this problem is a minimum eigenvalue maximization problem over a simple compact (a norm ball in fact), large-scale instances can be solved efficiently using projected gradient algorithms or smooth semidefinite optimization techniques [7,20]. As we show below, the complexity of projecting on this ball is quite low.

Lemma 5 *The complexity of projecting* $(x_0, y_0, z_0, w_0) \in \mathbf{R}^{3n}$ *on*

$$\|x\|_\infty + k^2\|y\|_\infty + \|z\|_1 + k\|w\|_\infty \leq \alpha$$

is bounded by $O(n \log n \log_2(1/\epsilon))$*, where* ϵ *is the target precision in projecting.*

Proof By duality, solving

$$\begin{array}{ll} \text{minimize} & \|x - x_0\|^2 + \|y - y_0\|^2 + \|z - z_0\|^2 + \|w - w_0\|^2 \\ \text{subject to} & \|x\|_\infty + k^2\|y\|_\infty + \|z\|_1 + k\|w\|_\infty \leq \alpha \end{array}$$

in the variables $x, y, z \in \mathbf{R}^n$ is equivalent to solving

$$\max_{\lambda \geq 0} \min_{x,y,z,w} \|(x, y, z, w) - (x_0, y_0, z_0, w_0)\|^2 + \lambda\|x\|_\infty + \lambda k^2\|y\|_\infty + \lambda\|z\|_1 + \lambda k\|w\|_\infty - \lambda\alpha$$

in the variable $\lambda \geq 0$. For a fixed λ, we can get the derivative w.r.t. λ by solving four separate penalized least-squares problems. Each of these problems can be solved explicitly in at most $O(n \log n)$ (by shrinking the current point) so the complexity of solving the outer maximization problem up to a precision $\epsilon > 0$ by binary search is $O(n \log n \log_2(1/\epsilon))$.

We can then implement the smooth minimization algorithm detailed in [19, Sect. 5.3] to a smooth approximation of problem (21) as in [20] or [7] for example. Let $\mu > 0$ be a regularization parameter. The function

$$f_\mu(X) = \mu \log\left(\mathbf{Tr} \exp\left(\frac{X}{\mu}\right)\right) \tag{22}$$

satifies

$$\lambda_{\max}(X) \leq f_\mu(X) \leq \lambda_{\max}(X) + \mu \log n$$

for any $X \in \mathbf{S}_n$. Furthermore, $f_\mu(X)$ is a smooth approximation of the function $\lambda_{\max}(X)$, and $\nabla f_\mu(X)$ is Lipschitz continuous with constant $\log n/\mu$. Let $\epsilon > 0$ be a given target precision, this means that if we set $\mu = \epsilon/(2\log n)$ then

$$f(U) \equiv -f_\mu\begin{pmatrix} -P^T U_1 P & \frac{1}{2}P^T(\mathbf{I} + U_4) \\ \frac{1}{2}(\mathbf{I} + U_4^T)P & -(U_2 + U_3) \end{pmatrix} \text{ where } U = (U_1, U_2, U_3, U_4), \tag{23}$$

will be an $\epsilon/2$ approximation of the objective function in (21). Whenever $\|U\|_F \leq 1$, we must have

$$\left\|\begin{pmatrix} -P^T U_1 P & P^T U_4/2 \\ U_4^T P/2 & -(U_2 + U_3) \end{pmatrix}\right\|_2^2 \leq \|P^T U_1 P\|_2^2 + \|U_2 + U_3\|_2^2 + \|P^T U_4\|_2^2 \leq 4,$$

hence, following [20, §4], the gradient of $f(U)$ is Lipschitz continuous with respect to the Frobenius norm, with Lipschitz constant given by

$$L = \frac{8\log(n+p)}{\epsilon},$$

We then define the compact, convex set Q as

$$Q \equiv \left\{(U_1, U_2, U_3, U_4) \in \mathbf{S}_n^3 : \|U_1\|_\infty + k^2\|U_2\|_\infty + \|U_3\|_1 + k\|U_4\|_\infty \le \bar{\alpha}\right\},$$

and define a prox function $d(U)$ over Q as $d(U) = \|U\|_F^2/2$, which is strongly convex with constant $\sigma = 1$ w.r.t. the Frobenius norm. Starting from $U_0 = 0$, the algorithm in [19] for solving

$$\begin{array}{ll} \text{maximize} & f(U) \\ \text{subject to} & U \in Q, \end{array}$$

where $f(U)$ is defined in (23), proceeds as follows.

```
**Repeat:**
  1. Compute $f(U_j)$ and $\nabla f(U_j)$
  2. Find $Y_j = \arg\min_{Y \in Q} \langle \nabla f(U_j), Y \rangle + \frac{1}{2} L \|U_i - Y\|_F^2$
  3. Find $W_j = \arg\min_{W \in Q} \left\{ \frac{Ld(W)}{\sigma} + \sum_{j=0}^{i} \frac{j+1}{2} (f(U_j) + \langle \nabla f(U_j), W - U_j \rangle) \right\}$
  4. Set $U_{j+1} = \frac{2}{j+3} W_j + \frac{j+1}{j+3} Y_j$
**Until** gap $\le \epsilon$.
```

Step one above computes the (smooth) function value and gradient. The second step computes the *gradient mapping*, which matches the gradient step for unconstrained problems (see [18, p. 86]). Step three and four update an *estimate sequence* see [18, p. 72] of f whose minimum can be computed explicitly and gives an increasingly tight upper bound on the minimum of f. We now present these steps in detail for our problem.

*Step*1 The most expensive step in the algorithm is the first, the computation of f and its gradient. This amounts to computing the matrix exponential in (22) at a cost of $O(n^3)$ (see [17] for details).

*Step*2 This step involves solving a problem of the form

$$\underset{Y \in Q}{\text{argmin}} \ \langle \nabla f(U), Y \rangle + \frac{1}{2} L \|U - Y\|_F^2,$$

where U is given. The above problem can be reduced to an Euclidean projection on Q

$$\underset{\|Y\| \in Q}{\text{argmin}} \ \|Y - V\|_F, \tag{24}$$

where $V = U + L^{-1}\nabla f_\mu(U)$ is given. According to Lemma 5, this can be solved $O(n \log n \log_2(1/\epsilon))$ opearations.

*Step*3 The third step involves solving an Euclidean projection problem similar to (24), with V defined here by:

$$V = \frac{\sigma}{L}\sum_{j=0}^{i}\frac{j+1}{2}\nabla f_\mu(U_j).$$

Stopping criterion We stop the algorithm when the duality gap is smaller than the target precision ϵ. The dual of the binary optimization problem (21) can be written

$$\begin{array}{ll}\text{minimize } \bar{\alpha}\max\{\|PG_{11}P^T\|_1, \frac{\|G_{22}\|_1}{k^2}, \|G_{22}\|_\infty, \frac{\|PG_{12}\|_1}{k}\} - \mathbf{Tr}(PG_{12}) \\ \text{subject to } \mathbf{Tr}(G) = 1,\ G \succeq 0,\end{array} \tag{25}$$

in the block matrix variable $G \in \mathbf{S}_{n+p}$ with blocks $G_{ij}, i, j = 1, 2$. Since the gradient $\nabla f(U)$ produces a dual feasible point by construction, we can use it to compute a dual objective value and bound the duality gap at the current point U.

Complexity According to [20], the total worst-case complexity to solve (21) with absolute accuracy less than ϵ is then given by

$$O\left(\frac{n^4\sqrt{\log n}}{\epsilon}\right)$$

Each iteration of the algorithm requires computing a matrix exponential at a cost of $O(n^3)$ and the algorithm requires $O(n\sqrt{\log n}/\epsilon)$ iterations to reach a target precision of $\epsilon > 0$. Note that while this smooth optimization method can be used to produce reasonable complexity bounds for checking if the optimal value of (21) is positive, i.e. if $\alpha_k \leq \bar{\alpha}$, in practice the algorithm is relatively slow and we mostly use interior point solvers on smaller problems to conduct experiments in the next section.

6 Numerical results

In this section, we illustrate the numerical performance of the semidefinite relaxation detailed in Sect. 3.

6.1 Illustration

We test the semidefinite relaxation in (11) on a sample of ten random Gaussian matrices $A \in \mathbf{R}^{p\times n}$ with $A_{ij} \sim \mathcal{N}(0, 1/\sqrt{p})$, $n = 30$ and $p = 22$. For each of these matrices, we solve problem (11) for $k = 2, \ldots, 5$ to produce upper bounds on α_k, hence on C_k in (5), with $\alpha_k = 1 - 1/C_k$. From [9], we know that if $\alpha_k < 1$ then we can bound the decoding error in (4), and if $\alpha_k < 1/2$ then the original signal can be recovered

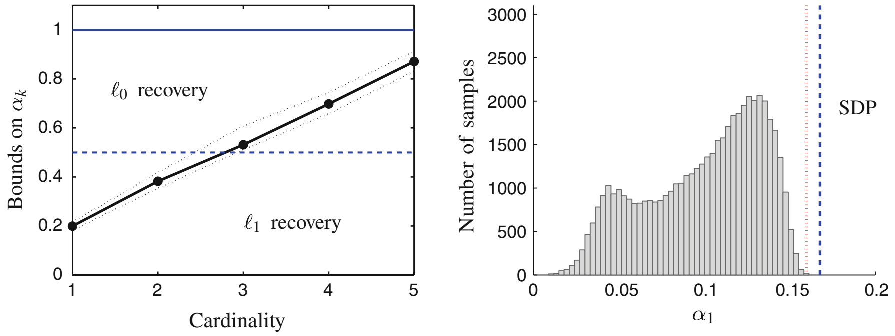

Fig. 1 Bounds on α_k. *Left* Upper bounds on α_k obtained by solving (11) for various values of k. Median bound over ten samples (*solid line*), *dotted lines* at pointwise minimum and maximum. *Right* Lower bound on α_1 obtained by randomization (*red dotted line*) compared with semidefinite relaxation bound (SDP *dashed line*)

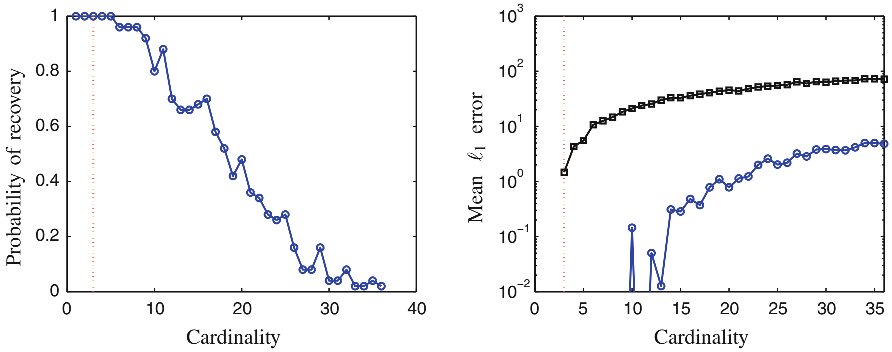

Fig. 2 Sparse recovery. *Left* Empirical probability of recovering the original sparse signal using the LP decoder in (3). The *dashed line* is at the strong recovery threshold. *Right* Empirical mean ℓ_1 recovery error $\|x - x_0\|_1$ using the LP decoder (*circles*) compared with the bound induced by Theorem 2 (*squares*)

exactly by solving a linear program. We also plot the randomized values for $y^T x$ with $k = 1$ together with the semidefinite relaxation bound (Fig. 1).

Next, in Fig. 2, we use a Gaussian matrix $A \in \mathbf{R}^{p \times n}$ with $A_{ij} \sim \mathcal{N}(0, 1/\sqrt{p})$, $n = 36$ and $p = 27$ and, for each k, we sample fifty information vectors $v = Ax_0$ where x_0 is uniformly distributed and has cardinality k. On the left, we plot the probability of recovering the original sparse signal x_0 using the linear programming decoder in (3). On the right, we plot the mean ℓ_1 recovery error $\|x - x_0\|_1$ using the linear programming decoder in (3) and compare it with the bound induced by Theorem 2.

6.2 Performance on compressed sensing matrices

In Tables 1, 2 and 3, we compare the performance of the linear programming relaxation bound on α_k derived in [14] with that of the semidefinite programming bound detailed in Sect. 3. We test these bounds for various matrix shape ratios $\rho = m/n$, target cardinalities k on matrices with Fourier, Bernoulli or Gaussian coefficients using

Table 1 Given ten sample *Fourier* matrices of leading dimension $n = 40$, we list median upper bounds on the values of α_k for various cardinalities k and matrix shape ratios ρ, computed using the linear programming (LP) relaxation in [14] and the semidefinite relaxation (SDP) detailed in this paper

Relaxation	ρ	α_1	α_2	α_3	α_4	α_5	Upper bound
LP	0.5	**0.21**	**0.38**	0.57	0.82	0.98	2
SDP	0.5	**0.21**	**0.38**	0.57	0.82	0.98	2
SDP low.	0.5	0.05	0.10	0.16	0.24	0.32	2
LP	0.6	**0.16**	**0.31**	**0.46**	0.61	0.82	3
SDP	0.6	**0.16**	**0.31**	**0.46**	0.61	0.82	3
SDP low.	0.6	0.04	0.09	0.15	0.20	0.31	3
LP	0.7	**0.12**	**0.25**	**0.39**	0.50	0.62	4
SDP	0.7	**0.12**	**0.25**	**0.39**	0.50	0.62	4
SDP low.	0.7	0.04	0.09	0.14	0.18	0.22	4
LP	0.8	**0.10**	**0.20**	**0.30**	**0.38**	**0.48**	6
SDP	0.8	**0.10**	**0.20**	**0.30**	**0.38**	**0.48**	6
SDP low.	0.8	0.04	0.07	0.13	0.17	0.23	6

We also list the upper bound on strong recovery computed using sequential convex optimization and the lower bound on α_k obtained by randomization using the SDP solution (SDP low.). Values of α_k below 1/2, for which strong recovery is certified, are highlighted in bold

Table 2 Given ten sample *Gaussian* matrices of leading dimension $n = 40$, we list median upper bounds on the values of α_k for various cardinalities k and matrix shape ratios ρ, computed using the linear programming (LP) relaxation in [14] and the semidefinite relaxation (SDP) detailed in this paper

Relaxation	ρ	α_1	α_2	α_3	α_4	α_5	Strong k	Weak k
LP	0.5	**0.27**	**0.49**	0.67	0.83	0.97	2	11
SDP	0.5	**0.27**	**0.49**	0.65	0.81	0.94	2	11
SDP low.	0.5	0.27	0.31	0.33	0.32	0.35	2	11
LP	0.6	**0.22**	**0.41**	0.57	0.72	0.84	2	12
SDP	0.6	**0.22**	**0.41**	0.56	0.70	0.82	2	12
SDP low.	0.6	0.22	0.29	0.31	0.32	0.36	2	12
LP	0.7	**0.20**	**0.34**	**0.47**	0.60	0.71	3	14
SDP	0.7	**0.20**	**0.34**	**0.46**	0.59	0.70	3	14
SDP low.	0.7	0.20	0.27	0.31	0.35	0.38	3	14
LP	0.8	**0.15**	**0.26**	**0.37**	**0.48**	0.58	3	16
SDP	0.8	**0.15**	**0.26**	**0.37**	**0.48**	0.58	3	16
SDP low.	0.8	0.15	0.23	0.28	0.33	0.38	3	16

We also list the asymptotic upper bound on both strong and weak recovery computed in [10] and the lower bound on α_k obtained by randomization using the SDP solution (SDP low.). Values of α_k below 1/2, for which strong recovery is certified, are highlighted in bold

SDPT3 by [22] to solve problem (11). We show median bounds computed over ten sample matrices for each type, hence test a total of 600 different matrices. We compare these relaxation bounds with the upper bounds produced by sequential convex optimization as in [14, Sect. 4.1]. In the Gaussian case, we also compare these relaxation bounds with the asymptotic thresholds on strong and weak (high probability)

Table 3 Given ten sample *Bernoulli* matrices of leading dimension $n = 40$, we list median upper bounds on the values of α_k for various cardinalities k and matrix shape ratios ρ, computed using the linear programming (LP) relaxation in [14] and the semidefinite relaxation (SDP) detailed in this paper

Relaxation	ρ	α_1	α_2	α_3	α_4	α_5	Upper bound
LP	0.5	**0.25**	**0.45**	0.64	0.82	0.97	2
SDP	0.5	**0.25**	**0.45**	0.63	0.80	0.94	2
SDP low.	0.5	0.25	0.28	0.29	0.29	0.34	2
LP	0.6	**0.21**	**0.38**	0.55	0.69	0.83	3
SDP	0.6	**0.21**	**0.38**	0.54	0.68	0.81	3
SDP low.	0.6	0.21	0.26	0.29	0.33	0.34	3
LP	0.7	**0.17**	**0.32**	**0.46**	0.58	0.70	4
SDP	0.7	**0.17**	**0.32**	**0.46**	0.58	0.69	4
SDP low.	0.7	0.17	0.24	0.29	0.33	0.37	4
LP	0.8	**0.14**	**0.26**	**0.38**	**0.47**	0.57	5
SDP	0.8	**0.14**	**0.26**	**0.37**	**0.47**	0.57	5
SDP low.	0.8	0.14	0.21	0.27	0.33	0.38	5

We also list the upper bound on strong recovery computed using sequential convex optimization and the lower bound on α_k obtained by randomization using the SDP solution (SDP low.). Values of α_k below 1/2, for which strong recovery is certified, are highlighted in bold

recovery discussed in [10]. The semidefinite bounds on α_k always match with the LP bounds in [14] when $k = 1$ (both are tight), and are often smaller than LP bounds whenever k is greater than 1 on Gaussian or Bernoulli matrices. The semidefinite upper bound on α_k was smaller than the LP one in 563 out of the 600 matrices sampled here, with the difference ranging from 4e−2 to −9e−4. Of course, this semidefinite relaxation is significantly more expensive than the LP based one and that these experiments thus had to be performed on very small matrices.

6.3 Tightness

Section 4 shows that the tightness of the semidefinite relaxation is explicitly controlled by the following quantity

$$\mu = g(X, \delta)h(Y, n, k, \delta),$$

where g and h are defined in (17, 18) respectively. In Fig. 3, we plot the histogram of values of μ for all 600 sample matrices computed above, and plot the same histogram on a subset of these results where the target cardinality k was set to 1. We observe that while the relaxation performed quite well on most of these examples, the randomization bound on performance often gets very large whenever $k > 1$. This can probably be explained by the fact that we only control the mean in Lemma 3, not the quantile. We also notice that μ is highly concentrated when $k = 1$ on Gaussian and Bernoulli matrices (where the results in Tables 2 and 3 are tight), while the performance is markedly worse for Fourier matrices.

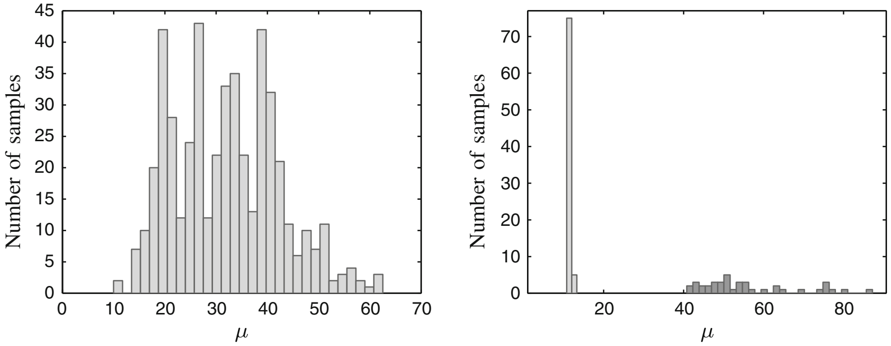

Fig. 3 Tightness. *Left* Histogram of $\mu = g(X, \delta)h(Y, n, k, \delta)$ defined in (17) and (18), computed for all sample solution matrices in the experiments above when $k > 1$. *Right* Idem using only examples where the target cardinality is $k = 1$, for Gaussian and Bernoulli matrices (*light grey*) or Fourier matrices (*dark grey*)

Finally, Tables 2 and 3 show that lower bounds on α_1 obtained by randomization for Gaussian are always tight (the solution of the SDP was very close to rank one), while performance on higher values of k and Fourier matrices is much worse. On 6 of these experiments however, the SDP randomization lower bound was higher than 1/2, which proved that $\alpha_5 > 1/2$, hence that the matrix did not satisfy the nullspace property at order 5.

6.4 Numerical complexity

We implemented the algorithm of Sect. (5) in MATLAB and tested it on random matrices. While the code handles matrices with $n = 500$, it is still considerably slower than similar first-order algorithms applied to sparse PCA problems for example (see [7]). A possible explanation for this gap in performance is perhaps that the DSPCA semidefinite relaxation is always tight (in practice at least) hence iterates near the solution tend to be very close to rank one. This is not the case here as the matrix in (9) is very rarely rank one and the number of significant eigenvalues has a direct impact on actual convergence speed. To illustrate this point, Fig. 4 shows a Scree plot of the optimal solution to (9) for a small Gaussian matrix (obtained by IP methods with a target precision of 10^{-8}), while Table 4 shows, as a benchmark, total CPU time for proving that $\alpha_1 < 1/2$ on Gaussian matrices, for various values of n. We set the accuracy $1e - 2$ and stop the code whenever positive objective values are reached. Unfortunately, performance for larger values of k is typically much worse (which is why we used IP methods to run most experiments in this section) and in many cases, convergence is hard to track as the dual objective values computed using the gradient in (25) produces a relatively coarse gap bounds as illustrated in Fig. 4 for a small Gaussian matrix.

Table 4 CPU time to show $\alpha_1 < 1/2$, using the algorithm of Sect. 5 on Gaussian matrices with shape ratio $\rho = 0.7$ for various values of n

n	50	100	200	500
CPU time	00 h 01 m	00 h 10 m	01 h 38 m	37 h 22 m

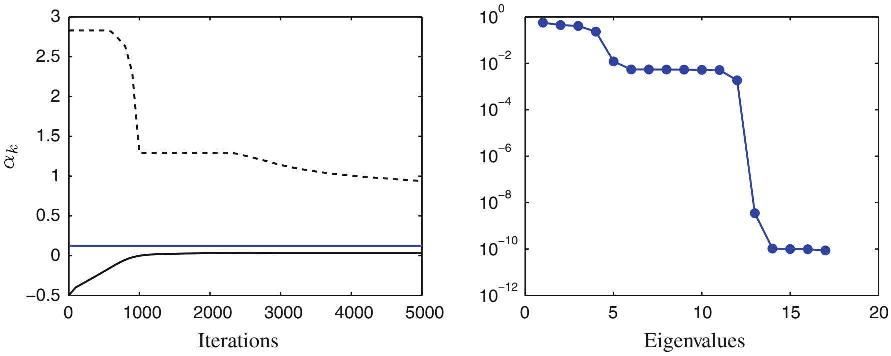

Fig. 4 Complexity. *Left* Primal and dual bounds on the optimal solution (computed using interior point methods) using the algorithm of Sect. 5 on a small Gaussian matrix. *Right* Scree plot of the optimal solution to (9) for a small Gaussian matrix (obtained by interior point methods with a target precision of 10^{-8})

7 Conclusion and directions for further research

We have detailed a semidefinite relaxation for the problem of testing if a matrix satisfies the nullspace property defined in [9] or [6]. This relaxation is tight for $k = 1$ and matches (numerically) the linear programming relaxation in [14]. It is often slightly tighter (again numerically) for larger values of k. We can also remark that the matrix A only appears in the relaxation (10) in "kernel" format $A^T A$, where the constraints are linear in the kernel matrix $A^T A$. This means that this relaxation might allow sparse experiment design problems to be solved, while maintaining convexity.

Of course, these small scale experiments do not really shed light on the actual performance of both relaxations on larger, more realistic problems. In particular, applications in imaging and signal processing would require solving problems where both n and k are several orders of magnitude larger than the values considered in this paper or in [14] and the question of finding tractable relaxations or algorithms that can handle such problem sizes remains open. Finally, the three different tractable tests for sparse recovery conditions, derived in [8,14] and this paper, are *all* limited to showing recovery at the (suboptimal) rate $k = O(\sqrt{m})$. Finding tractable test for sparse recovery at cardinalities k closer to the optimal rate $O(m)$ also remains an open problem.

Acknowledgments The authors are grateful to Arkadi Nemirovski (who suggested in particular the columnwise redundant constraints in (13) and the performance bounds) and Anatoli Juditsky for very helpful comments and suggestions. Would like to thank Ingrid Daubechies for first attracting our attention to the nullspace property. We are also grateful to two anonymous referees for numerous comments and suggestions. We thank Jared Tanner for forwarding us his numerical results. Finally, we acknowledge support from NSF grants DMS-0625352, SES-0835550 (CDI), CMMI-0844795 (CAREER) and CMMI-0968842, a Peek junior faculty fellowship and a Howard B. Wentz Jr. junior faculty award.

References

1. Affentranger, F., Schneider, R.: Random projections of regular simplices. Discrete Comput. Geom. **7**(1), 219–226 (1992)
2. Barvinok, A.: Integration and optimization of multivariate polynomials by restriction onto a random subspace. Foundations Comput. Math. **7**(2), 229–244 (2007)
3. Boyd, S., El Ghaoui, L., Feron, E., Balakrishnan, V.: Linear matrix inequalities in system and control theory. SIAM (1994)
4. Candès, E., Tao, T.: Near-optimal signal recovery from random projections: universal encoding strategies? IEEE Trans. Inform. Theory **52**(12), 5406–5425 (2006)
5. Candès, E.J., Tao, T.: Decoding by linear programming. IEEE Trans. Inform. Theory **51**(12), 4203–4215 (2005)
6. Cohen, A., Dahmen, W., DeVore, R.: Compressed sensing and best k-term approximation. J. AMS **22**(1), 211–231 (2009)
7. d'Aspremont, A., El Ghaoui, L., Jordan, M., Lanckriet, G.R.G.: A direct formulation for sparse PCA using semidefinite programming. SIAM Rev. **49**(3), 434–448 (2007)
8. d'Aspremont, A., Bach, F., El Ghaoui, L.: Optimal solutions for sparse principal component analysis. J. Mach. Learn. Res. **9**, 1269–1294 (2008)
9. Donoho, D., Huo, X.: Uncertainty principles and ideal atomic decomposition. IEEE Trans. Inform. Theory **47**(7), 2845–2862 (2001)
10. Donoho, D., Tanner, J.: Counting the faces of randomly-projected hypercubes and orthants, with applications. Arxiv Preprint **arXiv**, 08073590 (2008)
11. Donoho, D.L., Tanner, J.: Sparse nonnegative solutions of underdetermined linear equations by linear programming. Proc. Natl. Acad. Sci. **102**(27), 9446–9451 (2005)
12. Goemans, M., Williamson, D.: Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. J ACM **42**, 1115–1145 (1995)
13. Ibragimov, I., Sudakov, V., Tsirelson, B.: Norms of Gaussian sample functions. In: Proceedings of the third Japan USSR symposium on probability theory, lecture notes in math, vol. 550, pp. 20–41 (1976)
14. Juditsky, A., Nemirovski, A.: On verifiable sufficient conditions for sparse signal recovery via ℓ_1 minimization. ArXiv 08092650 (2008)
15. Lee, K., Bresler, Y.: Computing performance guarantees for compressed sensing. In: IEEE International Conference on Acoustics, Speech and Signal Processing, 2008. ICASSP 2008, pp 5129–5132 (2008)
16. Massart, P.: Concentration inequalities and model selection. Ecole d'Eté de Probabilités de Saint-Flour XXXIII (2007)
17. Moler, C., Van Loan, C.: Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later. SIAM Rev. **45**(1), 3–49 (2003)
18. Nesterov, Y.: Introductory Lectures on Convex Optimization. Springer, Heidelberg (2003)
19. Nesterov, Y.: Smooth minimization of non-smooth functions. Math. Program. **103**(1), 127–152 (2005)
20. Nesterov, Y.: Smoothing technique and its applications in semidefinite optimization. Math. Program. **110**(2), 245–259 (2007)
21. Sturm, J.: Using SEDUMI 1.0x, a MATLAB toolbox for optimization over symmetric cones. Optim. Method. Sofw. **11**, 625–653 (1999)
22. Toh, K.C., Todd, M.J., Tutuncu, R.H.: SDPT3—a MATLAB software package for semidefinite programming. Optim. Method. Sofw. **11**, 545–581 (1999)
23. Vershik, A., Sporyshev, P.: Asymptotic behavior of the number of faces of random polyhedra and the neighborliness problem. Selecta Math Soviet **11**(2), 181–201 (1992)
24. Zhang Y.: A simple proof for recoverability of ℓ_1-minimization: Go over or under. Rice University CAAM Technical report TR05-09 (2005)