Compressed Sensing: A Discrete Optimization Approach

Dimitris Bertsimas[1†] and Nicholas Johnson[2*†]

[1]Sloan School of Management, Massachusetts Institute of Technology, 100 Main Street, Cambridge, 02142, MA, USA.
[2]Operations Research Center, Massachusetts Institute of Technology, 1 Amherst Street, Cambridge, 02142, MA, USA.

*Corresponding author(s). E-mail(s): nagj@mit.edu;
Contributing authors: dbertsim@mit.edu;
[†]These authors contributed equally to this work.

Abstract

We study the Compressed Sensing (CS) problem, which is the problem of finding the most sparse vector that satisfies a set of linear measurements up to some numerical tolerance. CS is a central problem in Statistics, Operations Research and Machine Learning which arises in applications such as signal processing, data compression, image reconstruction, and multi-label learning. We introduce an ℓ_2 regularized formulation of CS which we reformulate as a mixed integer second order cone program. We derive a second order cone relaxation of this problem and show that under mild conditions on the regularization parameter, the resulting relaxation is equivalent to the well studied basis pursuit denoising problem. We present a semidefinite relaxation that strengthens the second order cone relaxation and develop a custom branch-and-bound algorithm that leverages our second order cone relaxation to solve small-scale instances of CS to certifiable optimality. When compared against solutions produced by three state of the art benchmark methods on synthetic data, our numerical results show that our approach produces solutions that are on average **6.22%** more sparse. When compared only against the experiment-wise best performing benchmark method on synthetic data, our approach produces solutions that are on average **3.10%** more sparse. On real world ECG data, for a given ℓ_2 reconstruction error our approach produces solutions that are on average **9.95%** more sparse than benchmark methods (**3.88%** more sparse if only compared against the best performing benchmark), while for a given sparsity level our approach produces solutions that have on average **10.77%** lower reconstruction error than benchmark methods (**1.42%** lower error if only compared against the best performing

benchmark). When used as a component of a multi-label classification algorithm, our approach achieves greater classification accuracy than benchmark compressed sensing methods. This improved accuracy comes at the cost of an increase in computation time by several orders of magnitude. Thus, for applications where runtime is not of critical importance, leveraging integer optimization can yield sparser and lower error solutions to CS than existing benchmarks.

Keywords: Sparsity; Sparse Approximation, Compressed Sensing; Convex Relaxation; Branch-and-bound

1 Introduction

The *Compressed Sensing* (CS) problem seeks to find a most sparse vector $\boldsymbol{x} \in \mathbb{R}^n$ that is consistent with a set of m linear equalities. CS is a fundamental problem in Statistics, Operations Research and Machine Learning which arises in numerous applications such as multi-label learning [1], medical resonance imaging [2], holography [3], climate monitoring [4], natural resource mining [5] and electrocardiogram signal acquisition [6] among many others. Formally, given a matrix $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ and a vector $\boldsymbol{b} \in \mathbb{R}^m$, CS is given by [7]:

$$\min_{\boldsymbol{x} \in \mathbb{R}^n} \|\boldsymbol{x}\|_0 \text{ s.t. } \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}. \tag{1}$$

In the presence of noisy measurements, it is necessary to relax the equality constraint in (1), leading to the following formulation for $\epsilon > 0$:

$$\min_{\boldsymbol{x} \in \mathbb{R}^n} \|\boldsymbol{x}\|_0 \text{ s.t. } \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2 \leq \epsilon. \tag{2}$$

This problem is sometimes referred to as sparse approximation in the literature [8] and trivially reduces to (1) for $\epsilon = 0$. CS allows signals to be reconstructed surprisingly well after sampling at a rate far below the Nyquist sampling rate by leveraging the inherent sparsity of most signals, either in the signal's latent space or in an appropriately defined transform space. For example, natural images tend to have a sparse representation in the wavelet domain, speech can be represented using a small number of coefficients in the Fourier transform domain and medical images can be represented sparsely in the Radon transform domain [9].

In Section 2, we will see that the vast majority of existing approaches to CS either rely on ℓ_1 based convex approximations to (2) or are greedy heuristics whereas the use of integer optimization techniques has gone relatively unexplored. In this work, we formulate CS as:

$$\min_{\boldsymbol{x} \in \mathbb{R}^n} \|\boldsymbol{x}\|_0 + \frac{1}{\gamma}\|\boldsymbol{x}\|_2^2 \text{ s.t. } \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2 \leq \epsilon, \tag{3}$$

where $\gamma > 0$ is a regularization parameter that in practice can either take a default value (e.g. $\sqrt{n}$) or be cross-validated by minimizing a validation metric [see, e.g., 10] to obtain strong out-of-sample performance [11]. A defining characteristic of the approach

we present in this work is that we leverage techniques from integer optimization to exploit the inherent discreteness of formulation (3) rather than relying on more commonly studied approximate methods. Note that Problem (3) is a special case of the formulation given by:

$$\min_{\boldsymbol{x} \in \mathbb{R}^n} \|\boldsymbol{x}\|_0 + \frac{1}{\gamma} \|\boldsymbol{W}\boldsymbol{x}\|_2^2 \text{ s.t. } \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2 \leq \epsilon. \tag{4}$$

where $\boldsymbol{W} \in \mathbb{R}^{n \times n}$ is a diagonal matrix with nonnegative diagonal entries that should be interpreted as coordinate weights on the vector $\boldsymbol{x}$. Indeed, (4) reduces to (3) when we take $\boldsymbol{W} = \boldsymbol{I}$.

1.1 Contributions and Structure

In this paper, we approach CS using mixed integer second order cone optimization. We derive a second order cone relaxation of this problem and show that under mild conditions on the regularization parameter, the resulting relaxation is equivalent to the well studied basis pursuit denoising problem. We present a semidefinite relaxation that strengthens the second order cone relaxation and develop a custom branch-and-bound algorithm that leverages our second order cone relaxation to solve instances of CS to certifiable optimality. Our numerical results show that our approach produces solutions that are on average 6.22% more sparse than solutions returned by three state of the art benchmark methods on synthetic data in minutes. If we restrict the comparison to the best performing benchmark method on each problem instance, our approach produces solutions that are on average 3.10% more sparse. On real world ECG data, for a given ℓ_2 reconstruction error our approach produces solutions that are on average 9.95% more sparse than benchmark methods (3.88% more sparse if only compared against the best performing benchmark), while for a given sparsity level our approach produces solutions that have on average 10.77% lower reconstruction error than benchmark methods in minutes (1.42% lower error if only compared against the best performing benchmark). On a real world multi-label classification task, our approach outperforms existing approaches in terms of accuracy, precision and recall. This increase in accuracy, precision and recall comes at the expense of a significant increase in running time of several orders of magnitude. Thus, for applications where runtime is not of critical importance, leveraging integer optimization can yield sparser and lower error solutions to CS than existing benchmarks.

The rest of the paper is structured as follows. In Section 2, we review existing formulations and solution methods of the CS problem. In Section 3, we study how our regularized formulation of CS (3) connects to the commonly used formulation (2). We reformulate (3) exactly as a mixed integer second order cone problem in Section 4 and present a second order cone relaxation in Section 4.1 and a stronger but more computationally expensive semidefinite cone relaxation in Section 4.2. We show that our second order cone relaxation is equivalent to the Basis Pursuit Denoising problem under mild conditions offering a new interpretation of this well studied method as a convex relaxation of our mixed integer second order cone reformulation of (3). We

3

leverage our second order cone relaxation to develop a custom branch-and-bound algorithm in Section 5 that can solve instances of (3) to certifiable optimality. In Section 6, we investigate the performance of our branch-and-bound algorithm against state of the art benchmark methods on synthetic data, real world ECG signal acquisition and real world multi-label classification.

Notation:

We let nonbold face characters such as b denote scalars, lowercase bold faced characters such as $\boldsymbol{x}$ denote vectors, uppercase bold faced characters such as $\boldsymbol{X}$ denote matrices, and calligraphic uppercase characters such as $\mathcal{Z}$ denote sets. We let $[n]$ denote the set of running indices $\{1, \ldots, n\}$. We let $\mathbf{1}_n$ denote an n-dimensional vector of all 1's, $\mathbf{0}_n$ denote an n-dimensional vector of all 0's, and $\boldsymbol{I}$ denote the identity matrix. We let $\mathcal{S}^n$ denote the subspace of $n \times n$ symmetric matrices and $\mathcal{S}^n_+ \in \mathcal{S}^n$ denote the cone of symmetric $n \times n$ positive semidefinite matrices.

2 Literature Review

In this section, we review several key approaches from the literature that have been employed to solve the CS problem. As an exhaustive literature review is outside of the scope of this paper, we focus our review on a handful of well studied approaches which will be used as benchmarks in this work. For a more detailed CS literature review, we refer the reader to [8].

The majority of existing approaches to the CS problem are heuristic in nature and generally can be classified as either convex approximations or greedy methods as we will see in this section. For these methods, associated performance guarantees require making strong statistical assumptions on the underlying problem data. Integer optimization has been given little attention in the CS literature despite its powerful modelling capabilities. [12] and [13] explore formulating Problem (2) as a mixed integer linear program for the case when $\epsilon = 0$. However this approach relies on using the big-M method which requires estimating reasonable values for M and cannot immediately generalize to the setting where $\epsilon > 0$.

2.1 Basis Pursuit Denoising

A common class of CS methods rely on solving convex approximations of (2) rather than attempting to solve (2) directly. A popular approach is to use the ℓ_1 norm as a convex surrogate for the ℓ_0 norm [7, 14–17]. This approximation is typically motivated by the observation that the unit ℓ_1 ball given by $\mathcal{B}_{\ell_1} = \{\boldsymbol{x} \in \mathbb{R}^n : \|\boldsymbol{x}\|_1 \leq 1\}$ is the convex hull of the nonconvex set $\mathcal{X} = \{\boldsymbol{x} \in \mathbb{R}^n : \|\boldsymbol{x}\|_0 \leq 1, \|\boldsymbol{x}\|_\infty \leq 1\}$. Replacing the ℓ_0 norm by the ℓ_1 norm in (2), we obtain:

$$\min_{\boldsymbol{x} \in \mathbb{R}^n} \|\boldsymbol{x}\|_1 \text{ s.t. } \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2 \leq \epsilon. \tag{5}$$

Problem (5) is referred to as Basis Pursuit Denoising and is a quadratically constrained convex optimization problem which can be solved efficiently using one of several off the

shelf optimization packages. Basis Pursuit Denoising produces an approximate solution to Problem (2) by either directly returning the solution of (5) or by post-processing the solution of (5) to further sparsify the result. One such post-processing technique is a greedy rounding mechanism where columns of the matrix $\boldsymbol{A}$ are iteratively selected in the order corresponding to decreasing magnitude of the entries of the optimal solution of (5) until the selected column set of $\boldsymbol{A}$ is sufficiently large to produce a feasible solution to (2). Basis Pursuit Denoising is very closely related to the Lasso problem which is given by:

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} \quad \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2 + \lambda\|\boldsymbol{x}\|_1, \tag{6}$$

where $\lambda > 0$ is a tunable hyperparameter. Lasso is a statistical estimator commonly used for sparse regression as empirically, the optimal solution of Problem (6) tends to be sparse [18]. More recently, strong connections between Lasso and robust optimization have been established [19]. Basis Pursuit Denoising and Lasso are equivalent in that Lasso is obtained by relaxing the hard constraint in (5) and instead introducing a penalty term in the objective function. It is straightforward to show that for given input data $\boldsymbol{A}, \boldsymbol{b}$ and ϵ in (5), there exists a value $\lambda^\star > 0$ such that there exists a solution $\boldsymbol{x}^\star$ that is both optimal for (5) and (6) when the tunable parameter takes value $\lambda = \lambda^\star$.

Note that by taking $\epsilon = 0$, Problem (5) reduces to the well studied Basis Pursuit problem where the equality constraint $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ is enforced. A large body of work studies conditions under which the optimal solution of the Basis Pursuit problem is also an optimal solution of (1). For example, see [20], [21], [22], and [23]. One of the most well studied conditions under which this equivalence holds is when the input matrix $\boldsymbol{A}$ satisfies the Restricted Isometry Property (RIP). Formally, a matrix $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ is said to satisfy RIP of order s and parameter $\delta_s \in (0, 1)$ if for every vector $\boldsymbol{x} \in \mathbb{R}^n$ such that $\|\boldsymbol{x}\|_0 \leq s$, we have

$$(1 - \delta_s)\|\boldsymbol{x}\|_2^2 \leq \|\boldsymbol{A}\boldsymbol{x}\|_2^2 \leq (1 + \delta_s)\|\boldsymbol{x}\|_2^2.$$

It has been established that if $\boldsymbol{A}$ satisfies RIP of order $2s$ and parameter $\delta_{2s} < 1/3$, then the optimal solution of the Basis Pursuit problem is also an optimal solution of (1) where s denotes the cardinality of this optimal solution [24]. While it has been shown that certain random matrices satisfy this desired RIP property with high probability [25, 26], RIP in general is not tractable to verify on arbitrary real world data.

2.2 Iterative Reweighted L1

Iterative Reweighted ℓ_1 minimization is an iterative method that can generate an approximate solution to (2) by solving a sequence of convex optimization problems that are very closely related to the Basis Pursuit Denoising problem given by (5) [27–29]. This approach falls in the class of convex approximation based methods for solving CS. The approach considers the weighted ℓ_1 minimization problem given by:

$$\begin{aligned} \min_{\boldsymbol{x}\in\mathbb{R}^n} \quad & \|\boldsymbol{W}\boldsymbol{x}\|_1 \\ \text{s.t.} \quad & \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2 \leq \epsilon, \end{aligned} \tag{7}$$

5

where $\boldsymbol{W} \in \mathbb{R}^{n \times n}$ is a diagonal matrix with nonnegative diagonal entries. Each diagonal entry $W_{ii} = w_i$ of $\boldsymbol{W}$ can be interpreted as a weighting of the ith coordinate of the vector $\boldsymbol{x}$. Interpreting the ℓ_1 norm as a convex surrogate for the ℓ_0 norm, Problem (7) can be viewed as a relaxation of the nonconvex problem given by

$$
\begin{aligned}
\min_{\boldsymbol{x} \in \mathbb{R}^n} \quad & \|\boldsymbol{W}\boldsymbol{x}\|_0 \\
\text{s.t.} \quad & \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2 \leq \epsilon.
\end{aligned} \tag{8}
$$

It is trivial to verify that when $\boldsymbol{W} = \alpha \boldsymbol{I}$, where $\alpha > 0$ and $\boldsymbol{I}$ is the n-by-n identity matrix, (8) and (7) reduce exactly to (2) and (5) respectively. Assuming the weights never vanish, the nonconvex Problems (2) and (8) have the same optimal solution, yet their convex relaxations (5) and (7) will generally have very different solutions. In this regard, the weights can be regarded as parameters that if chosen correctly can produce a better solution than (5). Iterative Reweighted ℓ_1 minimization proceeds as follows [27]:

1. Initialize the iteration count $t \leftarrow 0$ and the weights $w_i^{(0)} \leftarrow 1$.
2. Solve (7) with $\boldsymbol{W} = \boldsymbol{W}^{(t)}$. Let $\boldsymbol{x}^{(t)}$ denote the optimal solution.
3. Update the weights as $w_i^{(t+1)} \leftarrow \frac{1}{|x_i^{(t)}| + \delta}$ where $\delta > 0$ is a fixed parameter for numerical stability.
4. Terminate if t reaches a maximum number of iterations or if the iterates $\boldsymbol{x}^{(t)}$ have converged. Otherwise, increment t and return to Step 2.

It has been shown empirically that in many settings the solution returned by Iterated Reweighted ℓ_1 minimization outperforms the solution returned by Basis Pursuit Denoising by recovering the true underlying signal while requiring fewer measurements to be taken [27]. We note that this approach is an instance of a broader class of sparsifying iterative reweighted methods [30–32].

2.3 Orthogonal Matching Pursuit

Orthogonal Matching Pursuit (OMP) is a canonical greedy algorithm for obtaining heuristic solutions to (2) [33, 34]. Solving Problem (2) can be interpreted as determining the minimum number of columns from the input matrix $\boldsymbol{A}$ that must be selected such that the residual of the projection of the input vector $\boldsymbol{b}$ onto the span of the selected columns has ℓ_2 norm equal to at most $\sqrt{\epsilon}$. The OMP algorithm proceeds by first selecting the column of $\boldsymbol{A}$ that is most collinear with $\boldsymbol{b}$ and subsequently iteratively adding the column of $\boldsymbol{A}$ that is most collinear with the residual of the projection of $\boldsymbol{b}$ onto the subspace spanned by the selected columns until the norm of this residual is at most $\sqrt{\epsilon}$. Concretely, OMP proceeds as follows where for an arbitrary collection of indices $\mathcal{I}_t \subseteq [n]$, we let $\boldsymbol{A}(\mathcal{I}_t) \in \mathbb{R}^{m \times |\mathcal{I}_t|}$ denote the matrix obtained by stacking the $|\mathcal{I}_t|$ columns of $\boldsymbol{A}$ corresponding to the indices in the set $\mathcal{I}_t$:

1. Initialize the iteration count $t \leftarrow 0$, the residual $\boldsymbol{r}_0 \leftarrow \boldsymbol{b}$ and the index set $\mathcal{I}_0 \leftarrow \emptyset$.
2. Select the column that is most collinear with the residual $i_t \leftarrow \arg\max_{i \in [n] \setminus \mathcal{I}_t} |\boldsymbol{a}_i^T \boldsymbol{r}_t|$ and update the index set $\mathcal{I}_{t+1} \leftarrow \mathcal{I}_t \cup i_t$.

3. Compute the projection of $\boldsymbol{b}$ onto the current set of columns

$$\boldsymbol{x}_{t+1} \leftarrow \left[\boldsymbol{A}(\mathcal{I}_{t+1})^T \boldsymbol{A}(\mathcal{I}_{t+1})\right]^\dagger \boldsymbol{A}(\mathcal{I}_{t+1})^T \boldsymbol{b},$$

and update the residual $\boldsymbol{r}_{t+1} \leftarrow \boldsymbol{b} - \boldsymbol{A}(\mathcal{I}_{t+1})\boldsymbol{x}_{t+1}$.
4. Terminate if $\|\boldsymbol{r}_{t+1}\|_2^2 \leq \epsilon$, otherwise increment t and return to Step 2.

Conditions under which the solution returned by OMP is the optimal solution of (2) (either with high probability or with certainty) have been studied extensively [23, 35, 36]. Unfortunately, these conditions suffer from the same limitation as RIP in that in general they are not tractable to verify on real world data. A closely related method to OMP is Subspace Pursuit (SP) which is another greedy algorithm for obtaining a heuristic solution to (2) in the $\epsilon = 0$ setting but has the additional requirement that a target sparsity value K must be specified in advance [37]. SP is initialized by selecting the K columns of $\boldsymbol{A}$ that are most collinear with the vector $\boldsymbol{b}$. At each iteration, SP first computes the residual of the projection of $\boldsymbol{b}$ onto the current column set and then greedily updates up to K elements of the column set, repeating this process until doing so no longer decreases the norm of the residual.

3 Formulation Properties

In this section, we rigorously investigate connections between formulations (3) and (2) for the CS problem in the noisy setting. The only difference between formulations (2) and (3) is the inclusion of a ℓ_2 regularization term in the objective function in (3). We will see in Section 4 that the presence of this regularization term facilitates useful reformulations. Moreover, in the case of regression, [19] show that augmenting the ordinary least squares objective function with a ℓ_2 regularization penalty produces regression vectors that are robust against data perturbations which suggests the presence of such a regularization term may result in a similar benefit in (3). A natural question to ask is: under what conditions do problems (2) and (3) have the same solution? We answer this question in Theorem 1.

Theorem 1. *There exists a finite value $\gamma_0 < \infty$ such that for all $\bar{\gamma} \geq \gamma_0$, there exists a vector $\boldsymbol{x}^\star$ such that $\boldsymbol{x}^\star$ is an optimal solution of (2) and also an optimal solution of (3) where we set $\gamma = \bar{\gamma}$. Letting $\tilde{\boldsymbol{x}}$ denote a minimum norm solution to (2), we can take $\gamma_0 = \|\tilde{\boldsymbol{x}}\|_2^2$ and $\boldsymbol{x}^\star = \tilde{\boldsymbol{x}}$.*

Phrased simply, Theorem 1 establishes that there exists a finite value γ_0 such that if the regularization parameter γ in problem (3) is at least as large as γ_0, then there is a vector $\boldsymbol{x}^\star$ that is optimal to both problems (2) and (3). We note that this finite value γ_0 depends on the input data $\boldsymbol{A}, \boldsymbol{b}$ and ϵ.

Proof. Consider any matrix $\boldsymbol{A} \in \mathbb{R}^{m \times n}$, vector $\boldsymbol{b} \in \mathbb{R}^m$ and scalar $\epsilon > 0$. Let Ω denote the set of optimal solutions to (2) and let $\mathcal{X}$ denote the feasible set of (2) and (3). We have $\mathcal{X} = \{\boldsymbol{x} : \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2 \leq \epsilon\}$ and $\Omega \subseteq \mathcal{X}$. Let $\tilde{\boldsymbol{x}} \in \arg\min_{\boldsymbol{x} \in \Omega} \|\boldsymbol{x}\|_2^2$ and let $\gamma_0 = \|\tilde{\boldsymbol{x}}\|_2^2$. Since $\tilde{\boldsymbol{x}} \in \Omega$, $\tilde{\boldsymbol{x}}$ is an optimal solution to (2). It remains to show that $\tilde{\boldsymbol{x}}$ is optimal to (3) for all $\gamma \geq \gamma_0$.

Fix any $\gamma \geq \gamma_0$. To show that $\tilde{\boldsymbol{x}}$ is an optimal solution of (3), we will show that for all $\bar{\boldsymbol{x}} \in \mathcal{X}$, we have

$$\|\tilde{\boldsymbol{x}}\|_0 + \frac{1}{\gamma}\|\tilde{\boldsymbol{x}}\|_2^2 \leq \|\bar{\boldsymbol{x}}\|_0 + \frac{1}{\gamma}\|\bar{\boldsymbol{x}}\|_2^2.$$

Fix an arbitrary $\bar{\boldsymbol{x}} \in \mathcal{X}$. Either $\bar{\boldsymbol{x}} \in \mathcal{X} \setminus \Omega$ or $\bar{\boldsymbol{x}} \in \Omega$. Suppose $\bar{\boldsymbol{x}} \in \mathcal{X} \setminus \Omega$. The definition of Ω and the fact that $\tilde{\boldsymbol{x}} \in \Omega$ implies

$$\|\tilde{\boldsymbol{x}}\|_0 < \|\bar{\boldsymbol{x}}\|_0 \implies \|\tilde{\boldsymbol{x}}\|_0 + 1 \leq \|\bar{\boldsymbol{x}}\|_0.$$

Next, note that since $\gamma \geq \gamma_0 = \|\tilde{\boldsymbol{x}}\|_2^2$, we have

$$\|\tilde{\boldsymbol{x}}\|_0 + \frac{1}{\gamma}\|\tilde{\boldsymbol{x}}\|_2^2 \leq \|\tilde{\boldsymbol{x}}\|_0 + 1 \leq \|\bar{\boldsymbol{x}}\|_0 \leq \|\bar{\boldsymbol{x}}\|_0 + \frac{1}{\gamma}\|\bar{\boldsymbol{x}}\|_2^2.$$

Suppose instead that $\bar{\boldsymbol{x}} \in \Omega$. The definition of Ω and $\tilde{\boldsymbol{x}}$ imply $\|\tilde{\boldsymbol{x}}\|_0 = \|\bar{\boldsymbol{x}}\|_0$ and $\|\tilde{\boldsymbol{x}}\|_2^2 \leq \|\bar{\boldsymbol{x}}\|_2^2$. It then follows immediately that $\|\tilde{\boldsymbol{x}}\|_0 + \frac{1}{\gamma}\|\tilde{\boldsymbol{x}}\|_2^2 \leq \|\bar{\boldsymbol{x}}\|_0 + \frac{1}{\gamma}\|\bar{\boldsymbol{x}}\|_2^2$. Thus, $\tilde{\boldsymbol{x}}$ is optimal to (3). This completes the proof. $\square$

Though Theorem 1 is useful in establishing conditions for the equivalence of problems (2) and (3), it is important to note that computing the value of γ_0 specified in the Theorem requires solving (2) which is difficult in general. Suppose we are solving problem (3) with some regularization parameter γ in the regime where $0 < \gamma < \gamma_0$. A natural question to ask is: how well does the solution of (3) approximate the solution of (2). We answer this question in Theorem 2.

Theorem 2. *Let $\tilde{\boldsymbol{x}}$ and γ_0 be as defined in Theorem 1, and let $\mathcal{X}$ denote the feasible set of (2) and (3). Specifically, $\tilde{\boldsymbol{x}}$ denotes a minimum norm solution to (2), $\gamma_0 = \|\tilde{\boldsymbol{x}}\|_2^2$ and $\mathcal{X} = \{\boldsymbol{x} : \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2 \leq \epsilon\}$. Let $\lambda_\epsilon > 0$ be a value such that*

$$\arg\min_{\boldsymbol{x} \in \mathcal{X}} \|\boldsymbol{x}\|_2^2 = \arg\min_{\boldsymbol{x}} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2 + \lambda_\epsilon\|\boldsymbol{x}\|_2^2.$$

Fix any value γ with $0 < \gamma < \gamma_0$. Suppose $\bar{\boldsymbol{x}}$ is an optimal solution to (3). Then we have

$$\|\tilde{\boldsymbol{x}}\|_0 \leq \|\bar{\boldsymbol{x}}\|_0 \leq \|\tilde{\boldsymbol{x}}\|_0 + \frac{1}{\gamma}\left(\|\tilde{\boldsymbol{x}}\|_2^2 - \left\|\left(\frac{1}{\lambda_\epsilon}\boldsymbol{I} + \boldsymbol{A}^T\boldsymbol{A}\right)^{-1}\boldsymbol{A}^T\boldsymbol{b}\right\|_2^2\right).$$

Proof. Fix any value γ with $0 < \gamma < \gamma_0$ and consider any optimal solution $\bar{\boldsymbol{x}}$ to (3). The inequality $\|\tilde{\boldsymbol{x}}\|_0 \leq \|\bar{\boldsymbol{x}}\|_0$ follows immediately from the optimality of $\tilde{\boldsymbol{x}}$ in (2). By the optimality of $\bar{\boldsymbol{x}}$, we must have

$$\|\bar{\boldsymbol{x}}\|_0 + \frac{1}{\gamma}\|\bar{\boldsymbol{x}}\|_2^2 \leq \|\tilde{\boldsymbol{x}}\|_0 + \frac{1}{\gamma}\|\tilde{\boldsymbol{x}}\|_2^2 \implies \|\bar{\boldsymbol{x}}\|_0 \leq \|\tilde{\boldsymbol{x}}\|_0 + \frac{1}{\gamma}(\|\tilde{\boldsymbol{x}}\|_2^2 - \|\bar{\boldsymbol{x}}\|_2^2).$$

Thus, to establish the result we need only derive an upper bound for the term $(\|\tilde{\boldsymbol{x}}\|_2^2 - \|\bar{\boldsymbol{x}}\|_2^2)$, or equivalently to derive a lower bound for the term $\|\bar{\boldsymbol{x}}\|_2^2$. Since $\bar{\boldsymbol{x}} \in \mathcal{X}$, such

8

a lower bound can be obtained by solving the optimization problem given by

$$\min_{\boldsymbol{x} \in \mathbb{R}^n} \quad \|\boldsymbol{x}\|_2^2$$
$$\text{s.t.} \qquad \boldsymbol{x} \in \mathcal{X} = \{\boldsymbol{x} : \|\boldsymbol{Ax} - \boldsymbol{b}\|_2^2 \le \epsilon\}. \tag{9}$$

This optimization problem has the same optimal solution as the ridge regression problem given by

$$\min_{\boldsymbol{x} \in \mathbb{R}^n} \|\boldsymbol{Ax} - \boldsymbol{b}\|_2^2 + \lambda_\epsilon \|\boldsymbol{x}\|_2^2. \tag{10}$$

for some value $\lambda_\epsilon > 0$. To see this, we form the Lagrangian for (9) $L(\boldsymbol{x}, \mu) = \|\boldsymbol{x}\|_2^2 + \mu(\|\boldsymbol{Ax} - \boldsymbol{b}\|_2^2 - \epsilon)$ and observe that the KKT conditions for $(\boldsymbol{x}, \mu) \in \mathbb{R}^n \times \mathbb{R}$ are given by

1. $\|\boldsymbol{Ax} - \boldsymbol{b}\|_2^2 \le \epsilon$;
2. $\mu \ge 0$;
3. $\mu(\|\boldsymbol{Ax} - \boldsymbol{b}\|_2^2 - \epsilon) = 0 \implies \mu = 0$ or $\|\boldsymbol{Ax} - \boldsymbol{b}\|_2^2 = \epsilon$;
4. $\nabla_{\boldsymbol{x}} L(\boldsymbol{x}, \mu) = \boldsymbol{0} \implies \boldsymbol{x} = (\frac{1}{\mu} \boldsymbol{I} + \boldsymbol{A}^T \boldsymbol{A})^{-1} \boldsymbol{A}^T \boldsymbol{b}$ if $\mu \ne 0$ and $\boldsymbol{x} = 0$ if $\mu = 0$.

We note that if $\boldsymbol{0} \in \mathcal{X}$, then $\boldsymbol{0}$ is trivially an optimal solution to (9) with optimal value given by 0. This corresponds to the degenerate case. In the nondegenerate case, we have $\boldsymbol{0} \notin \mathcal{X}$. This condition, coupled with the first and fourth KKT conditions implies that at optimality, we have $\mu \ne 0$ and $\boldsymbol{x} = (\frac{1}{\mu} \boldsymbol{I} + \boldsymbol{A}^T \boldsymbol{A})^{-1} \boldsymbol{A}^T \boldsymbol{b}$. Next, we note that the unconstrained quadratic optimization problem given by (10) has an optimal solution $\boldsymbol{x}^\star$ given by $\boldsymbol{x}^\star = (\lambda_\epsilon \boldsymbol{I} + \boldsymbol{A}^T \boldsymbol{A})^{-1} \boldsymbol{A}^T \boldsymbol{b}$. Finally, we observe that the two preceding expressions are the same when $\lambda_\epsilon = \frac{1}{\mu} > 0$. Thus, we have

$$\|\bar{\boldsymbol{x}}\|_2^2 \ge \min_{\boldsymbol{x} \in \mathcal{X}} \|\boldsymbol{x}\|_2^2 = \left\| \left(\frac{1}{\lambda_\epsilon} \boldsymbol{I} + \boldsymbol{A}^T \boldsymbol{A} \right)^{-1} \boldsymbol{A}^T \boldsymbol{b} \right\|_2^2,$$

which implies that

$$\|\bar{\boldsymbol{x}}\|_0 \le \|\tilde{\boldsymbol{x}}\|_0 + \frac{1}{\gamma} \left(\|\tilde{\boldsymbol{x}}\|_2^2 - \left\| \left(\frac{1}{\lambda_\epsilon} \boldsymbol{I} + \boldsymbol{A}^T \boldsymbol{A} \right)^{-1} \boldsymbol{A}^T \boldsymbol{b} \right\|_2^2 \right).$$

This completes the proof. $\qquad\qquad\square$

Remark 1. *Though the statement of Theorem 2 is made for any fixed γ satisfying $0 < \gamma < \gamma_0$ with γ_0 given by Theorem 1, we note that the proof of Theorem 2 in fact generalizes to any $\gamma > 0$. This implies that the result of Theorem 1 holds for any γ_0' satisfying $\gamma_0' > \left(\|\tilde{\boldsymbol{x}}\|_2^2 - \left\| \left(\frac{1}{\lambda_\epsilon} \boldsymbol{I} + \boldsymbol{A}^T \boldsymbol{A} \right)^{-1} \boldsymbol{A}^T \boldsymbol{b} \right\|_2^2 \right)$. This is a stronger condition than the one established by Theorem 1 but has the drawback of depending on the value λ_ϵ which in general cannot be computed easily.*

Theorem 2 provides a worst case guarantee on the sparsity of the solution of (3) when the regularization parameter γ satisfies $0 < \gamma < \gamma_0$.

4 An Exact Reformulation and Convex Relaxations

In this section, we reformulate (4) as a mixed integer second order cone optimization problem. We then employ the perspective relaxation [38] to construct a second order cone relaxation for (4) and demonstrate that under certain conditions on the regularization parameter γ, the resulting relaxation is equivalent to the Weighted Basis Pursuit Denoising problem given by (7). As a special case, we obtain a convex relaxation for (3) and demonstrate that it is equivalent to (5) under the same conditions on γ. Finally, we present a family of semidefinite relaxations to (4) using techniques from polynomial optimization.

To model the sparsity of the vector $\boldsymbol{x}$ in (4), we introduce binary variables $\boldsymbol{z} \in \{0,1\}^n$ and require that $x_i = z_i x_i$. This gives the following reformulation of (4):

$$\min_{\boldsymbol{z}, \boldsymbol{x} \in \mathbb{R}^n} \quad \sum_{i=1}^n z_i + \frac{1}{\gamma} \sum_{i=1}^n w_i^2 x_i^2 \tag{11}$$
$$\text{s.t.} \quad \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2 \leq \epsilon, \ x_i = z_i x_i \ \forall \ i, \ z_i \in \{0,1\} \ \forall \ i.$$

The constraints $x_i = z_i x_i$ in (11) are nonconvex in the decision variables $(\boldsymbol{x}, \boldsymbol{z})$. To deal with these constraints, we make use of the perspective reformulation [38]. Specifically, we introduce non-negative variables $\boldsymbol{\theta} \in \mathbb{R}_+^n$ where θ_i models x_i^2 and introduce the constraints $\theta_i z_i \geq x_i^2$, which are second order cone representable. Thus, if $z_i = 0$, we will have $x_i = 0$. This results in the following reformulation of (11):

$$\min_{\boldsymbol{z}, \boldsymbol{x}, \boldsymbol{\theta} \in \mathbb{R}^n} \quad \sum_{i=1}^n z_i + \frac{1}{\gamma} \sum_{i=1}^n w_i^2 \theta_i \tag{12}$$
$$\text{s.t.} \quad \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2 \leq \epsilon, \ x_i^2 \leq z_i \theta_i \ \forall \ i,$$
$$z_i \in \{0,1\} \ \forall \ i, \ \theta_i \geq 0 \ \forall \ i.$$

Theorem 3. *The mixed integer second order cone problem given by* (12) *is an exact reformulation of* (4).

Proof. We show that given a feasible solution to (4), we can construct a feasible solution to (12) that achieves the same objective value and vice versa.

Consider an arbitrary solution $\bar{\boldsymbol{x}}$ to (4). Let $\bar{\boldsymbol{z}} \in \mathbb{R}^n$ be the binary vector obtained by setting $\bar{z}_i = \mathbb{1}\{\bar{x}_i \neq 0\}$ and let $\bar{\boldsymbol{\theta}} \in \mathbb{R}^n$ be the vector obtained by setting $\bar{\theta}_i = \bar{x}_i^2$. We have $\|\boldsymbol{A}\bar{\boldsymbol{x}} - \boldsymbol{b}\|_2^2 \leq \epsilon$, $\bar{z}_i \bar{\theta}_i = \mathbb{1}\{\bar{x}_i \neq 0\} \cdot \bar{x}_i^2 = \bar{x}_i^2$, $\bar{\boldsymbol{z}} \in \{0,1\}^n$ and $\bar{\theta}_i \geq 0$ so the solution $(\bar{\boldsymbol{x}}, \bar{\boldsymbol{z}}, \bar{\boldsymbol{\theta}})$ is feasible to (12). Lastly, notice that we have

$$\sum_{i=1}^n \bar{z}_i + \frac{1}{\gamma} \sum_{i=1}^n w_i^2 \bar{\theta}_i = \sum_{i=1}^n \mathbb{1}\{\bar{x}_i \neq 0\} + \frac{1}{\gamma} \sum_{i=1}^n w_i^2 \bar{x}_i^2 = \|\bar{\boldsymbol{x}}\|_0 + \frac{1}{\gamma} \|\boldsymbol{W}\bar{\boldsymbol{x}}\|_2^2,$$

where $\boldsymbol{W} = \text{diag}(w_1, \ldots, w_n)$. Thus, the solution $(\bar{\boldsymbol{x}}, \bar{\boldsymbol{z}}, \bar{\boldsymbol{\theta}})$ is a feasible solution to (12) that achieves the same objective value as $\bar{\boldsymbol{x}}$ does in (4).

Consider now an arbitrary solution $(\bar{\boldsymbol{x}}, \bar{\boldsymbol{z}}, \bar{\boldsymbol{\theta}})$ to (12). Since we have $\|\boldsymbol{A}\bar{\boldsymbol{x}} - \boldsymbol{b}\|_2^2 \leq \epsilon$, $\bar{\boldsymbol{x}}$ is feasible to (4). Next, we note that the constraints $x_i^2 \leq z_i \theta_i$ and $z_i \in \{0,1\}$ imply

that $\bar{z}_i \geq \mathbb{1}\{\bar{x}_i \neq 0\}$ and $\bar{\theta}_i \geq \bar{x}_i^2$. Finally, we observe that

$$\|\bar{\boldsymbol{x}}\|_0 + \frac{1}{\gamma}\|\boldsymbol{W}\bar{\boldsymbol{x}}\|_2^2 = \sum_{i=1}^n \mathbb{1}\{\bar{x}_i \neq 0\} + \frac{1}{\gamma}\sum_{i=1}^n w_i^2 \bar{x}_i^2 \leq \sum_{i=1}^n \bar{z}_i + \frac{1}{\gamma}\sum_{i=1}^n w_i^2 \bar{\theta}_i.$$

Thus, the solution $\bar{\boldsymbol{x}}$ is a feasible solution to (4) that achieves an objective value equal to or less than the objective value that $(\bar{\boldsymbol{x}}, \bar{\boldsymbol{z}}, \bar{\boldsymbol{\theta}})$ achieves in (12). This completes the proof. $\qquad\square$

4.1 A Second Order Cone Relaxation

Problem (12) is a reformulation of Problem (4) where the problem's nonconvexity is entirely captured by the binary variables $\boldsymbol{z}$. We now obtain a convex relaxation of (4) by solving (12) with $\boldsymbol{z} \in \text{conv}(\{0,1\}^n) = [0,1]^n$. This gives the following convex optimization problem:

$$
\begin{aligned}
\min_{\boldsymbol{z},\boldsymbol{x},\boldsymbol{\theta}\in\mathbb{R}^n} \quad & \sum_{i=1}^n z_i + \frac{1}{\gamma}\sum_{i=1}^n w_i^2 \theta_i \\
\text{s.t.} \quad & \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2 \leq \epsilon, \ x_i^2 \leq z_i \theta_i \ \forall \ i, \\
& 0 \leq z_i \leq 1 \ \forall \ i, \ \theta_i \geq 0 \ \forall \ i.
\end{aligned}
\tag{13}
$$

A natural question to ask is how problem (13) compares to the Weighted Basis Pursuit Denoising problem given by (7), a common convex approximation for CS in the noisy setting. Surprisingly, under mild conditions on the regularization parameter γ, it can be shown that solving (13) is exactly equivalent to solving (7). This implies that though Basis Pursuit Denoising is typically motivated as a convex approximation to CS in the presence of noise, it can alternatively be understood as the natural convex relaxation of the mixed integer second order cone problem given by (12) for appropriately chosen values of γ. We formalize this statement in Theorem 4.

Theorem 4. *There exists a finite value $\gamma_0 < \infty$ such that for all $\bar{\gamma} \geq \gamma_0$, any vector $\boldsymbol{x}^\star$ that is an optimal solution of (7) is also an optimal solution of (13). Let $\mathcal{X} = \{\boldsymbol{x} : \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2 \leq \epsilon\}$, the feasible set of (7). We can take $\gamma_0 = \max_{x\in\mathcal{X}}\|\boldsymbol{W}\boldsymbol{x}\|_\infty^2$, where $\boldsymbol{W} = diag(w_1,\ldots,w_n)$.*

Proof. Rewrite (13) as the two stage optimization problem given by (14).

$$
\begin{aligned}
\min_{\boldsymbol{x}\in\mathcal{X}} \quad \min_{\boldsymbol{z},\boldsymbol{\theta}\in\mathbb{R}^n} \quad & \sum_{i=1}^n z_i + \frac{1}{\gamma}\sum_{i=1}^n w_i^2 \theta_i \\
\text{s.t.} \quad & x_i^2 \leq z_i \theta_i \ \forall \ i, \ 0 \leq z_i \leq 1 \ \forall \ i, \ \theta_i \geq 0 \ \forall \ i.
\end{aligned}
\tag{14}
$$

Let $\gamma_0 = \max_{x\in\mathcal{X}}\|x\|_\infty^2$. To establish the result, we will show that for any $\boldsymbol{x} \in \mathcal{X}$ the optimal value of the inner minimization problem in (14) is a scalar multiple of the ℓ_1 norm of $\boldsymbol{W}\boldsymbol{x}$ provided that $\gamma \geq \gamma_0$.

Fix $\gamma \geq \gamma_0$ and consider any $\bar{\boldsymbol{x}} \in \mathcal{X}$. We make three observations that allow us to reformulate the inner minimization problem in (14):

1. The objective function of the inner minimization problem is separable.

11

2. For any i such that $\bar{x}_i = 0$, it is optimal to set $z_i = \theta_i = 0$ which results in no contribution to the objective function.

3. For any i such that $\bar{x}_i \neq 0$, we must have $z_i > 0$ and it is optimal to take $\theta_i = \frac{\bar{x}_i^2}{z_i}$.

We can therefore equivalently express the inner minimization problem of (14) as:

$$\min_{\boldsymbol{z} \in \mathbb{R}^n} \quad \sum_{i:x_i \neq 0} \left[z_i + \frac{w_i^2}{\gamma} \cdot \frac{\bar{x}_i^2}{z_i} \right] \tag{15}$$
$$\text{s.t.} \quad 0 < z_i \leq 1 \ \forall \ i.$$

Let $f_i(z) = z + \frac{w_i^2}{\gamma} \cdot \frac{\bar{x}_i^2}{z}$. We want to minimize the function $f_i(z)$ over the interval $(0,1]$ for all i such that $\bar{x}_i \neq 0$. Fix an arbitrary i satisfying $\bar{x}_i \neq 0$. We have $\frac{d}{dz} f_i(z) = 1 - \frac{w_i^2}{\gamma} \cdot \frac{\bar{x}_i^2}{z_i^2}$ and $\frac{d}{dz} f_i(z^\star) = 0 \iff z^\star = \pm \frac{w_i}{\sqrt{\gamma}} |\bar{x}_i|$. The condition $\gamma \geq \gamma_0 = \max_{x \in \mathcal{X}} \|\boldsymbol{W}\boldsymbol{x}\|_\infty^2$ and the fact that $\bar{\boldsymbol{x}} \in \mathcal{X}$ implies that $1 \geq \frac{w_i^2 \bar{x}_i^2}{\gamma}$ for all i. Thus, we have $0 < \frac{w_i}{\sqrt{\gamma}} |\bar{x}_i| \leq 1$. Let $\bar{z} = \frac{w_i}{\sqrt{\gamma}} |\bar{x}_i|$. Noting that $\lim_{z \to 0^+} f_i(z) = \infty$, the minimum of $f_i(z)$ over the interval $(0,1]$ must occur either at 1 or $\bar{z}$. We have

$$\left(\frac{w_i}{\sqrt{\gamma}} |\bar{x}_i| - 1 \right)^2 \geq 0 \implies f_i(1) = \frac{w_i^2 \bar{x}_i^2}{\gamma} + 1 \geq \frac{2 w_i}{\sqrt{\gamma}} |\bar{x}_i| = f_i(\bar{z}).$$

Therefore, the minimum of $f_i(z)$ on $(0,1]$ occurs at $\bar{z} = \frac{w_i}{\sqrt{\gamma}} |\bar{x}_i|$ and is equal to $f_i(\bar{z}) = \frac{2}{\sqrt{\gamma}} |\bar{x}_i|$. This allows us to conclude that the optimal value of (14) is given by:

$$\sum_{i:x_i \neq 0} \frac{2 w_i}{\sqrt{\gamma}} |\bar{x}_i| = \sum_{i=1}^n \frac{2 w_i}{\sqrt{\gamma}} |\bar{x}_i| = \frac{2}{\sqrt{\gamma}} \|\boldsymbol{W}\bar{\boldsymbol{x}}\|_1.$$

We have shown that for fixed $\boldsymbol{x} \in \mathcal{X}$, the optimal value of the inner minimization problem of (14) is a scalar multiple of the ℓ_1 norm of $\boldsymbol{W}\boldsymbol{x}$. We can rewrite (14) as

$$\min_{\boldsymbol{x} \in \mathcal{X}} \frac{2}{\sqrt{\gamma}} \|\boldsymbol{W}\bar{\boldsymbol{x}}\|_1, \tag{16}$$

which has the same set of optimal solutions as (7) because this set is invariant under scaling of the objective function. This completes the proof. $\qquad\square$

Remark 2. *Note that by taking $\boldsymbol{W} = \boldsymbol{I}$, it immediately follows from Theorem 4 that any vector $\boldsymbol{x}^\star$ that is an optimal solution of (5) is also an optimal solution of (13) when we set $\gamma \geq \gamma_0 = \max_{\boldsymbol{x} \in \mathcal{X}} \|\boldsymbol{x}\|_\infty^2$.*

Convex relaxations of nonconvex optimization problems are helpful for two reasons. Firstly, a convex relaxation provides a lower (upper) bound to a minimization (maximization) problem which given a feasible solution to the nonconvex optimization problem provides a certificate of worst case suboptimality. Secondly, convex relaxations can often be used as building blocks in the construction of global optimization

algorithms or heuristics for nonconvex optimization problems. Strong convex relaxations are desirable because they produce tighter bounds on the optimal value of the problem of interest (stronger certificates of worst case suboptimality) and generally lead to more performant global optimization algorithms and heuristics. Let $\mathcal{X}_1 = \{(\boldsymbol{z}, \boldsymbol{x}, \boldsymbol{\theta}) \in \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^n : \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2 \le \epsilon, x_i^2 \le z_i\theta_i \ \forall \ i, \theta_i \ge 0 \ \forall \ i\}$ and $\mathcal{X}_1 = \{(\boldsymbol{z}, \boldsymbol{x}, \boldsymbol{\theta}) \in \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^n : \boldsymbol{z} \in \{0, 1\}^n\}$. We can equivalently write (12) as:

$$\min_{(\boldsymbol{z}, \boldsymbol{x}, \boldsymbol{\theta}) \in \mathcal{X}_1 \cap \mathcal{X}_2} \sum_{i=1}^n z_i + \frac{1}{\gamma} \sum_{i=1}^n w_i^2 \theta_i.$$

The strongest possible convex relaxation to (12) would be obtained by minimizing the objective function in (12) subject to the constraint that $(\boldsymbol{z}, \boldsymbol{x}, \boldsymbol{\theta}) \in \mathrm{conv}(\mathcal{X}_1 \cap \mathcal{X}_2)$. Since the objective function is linear in the decision variables, solving over $\mathrm{conv}(\mathcal{X}_1 \cap \mathcal{X}_2)$ would produce an optimal solution to (12) since the objective would be minimized at an extreme point of $\mathrm{conv}(\mathcal{X}_1 \cap \mathcal{X}_2)$ which by definition must be an element of $\mathcal{X}_1 \cap \mathcal{X}_2$. Unfortunately, in general it is hard to represent $\mathrm{conv}(\mathcal{X}_1 \cap \mathcal{X}_2)$ explicitly. The relaxation given by (13) consists of minimizing the objective function of (12) subject to the constraint that $(\boldsymbol{z}, \boldsymbol{x}, \boldsymbol{\theta}) \in (\mathrm{conv}(\mathcal{X}_1) \cap \mathrm{conv}(\mathcal{X}_2)) = \mathcal{X}_1 \cap \mathrm{conv}(\mathcal{X}_2) \supseteq \mathrm{conv}(\mathcal{X}_1 \cap \mathcal{X}_2)$.

Stronger convex relaxations to (12) can be obtained by introducing additional valid inequalities to (12) and then relaxing the integrality constraint on $\boldsymbol{z}$. For example, suppose we know a value $M \ge \gamma_0 = \max_{\boldsymbol{x} \in \mathcal{X}} \|\boldsymbol{W}\boldsymbol{x}\|_\infty^2$. We can use this value to introduce Big-M constraints similar in flavour to the formulation proposed by [12]. Under this assumption, it follows immediately that any feasible solution to (12) satisfies $-Mz_i \le w_i x_i \le Mz_i \ \forall \ i$. Thus, we can obtain another convex relaxtion of (12) by minimizing its objective function subject to the constraint $(\boldsymbol{z}, \boldsymbol{x}, \boldsymbol{\theta}) \in \bar{\mathcal{X}}_1 \cap \mathrm{conv}(\mathcal{X}_2) \supseteq \mathrm{conv}(\mathcal{X}_1 \cap \mathcal{X}_2)$ where we define $\bar{\mathcal{X}}_1 = \mathcal{X}_1 \cap \{(\boldsymbol{z}, \boldsymbol{x}, \boldsymbol{\theta}) \in \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^n : -Mz_i \le w_i x_i \le Mz_i \ \forall \ i\}$. Explicitly, with knowledge of such a value M we can solve

$$\begin{aligned} \min_{\boldsymbol{z}, \boldsymbol{x}, \boldsymbol{\theta} \in \mathbb{R}^n} \quad & \sum_{i=1}^n z_i + \frac{1}{\gamma} \sum_{i=1}^n w_i^2 \theta_i \\ \text{s.t.} \quad & \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2 \le \epsilon, \ x_i^2 \le z_i\theta_i \ \forall \ i, \\ & -Mz_i \le w_i x_i \le Mz_i \ \forall \ i, \ 0 \le z_i \le 1 \ \forall \ i, \ \theta_i \ge 0 \ \forall \ i. \end{aligned} \quad (17)$$

Remark 3. *Given any input data $\boldsymbol{A}, \boldsymbol{b}, \epsilon$, if M satisfies $M \ge \gamma_0 = \max_{\boldsymbol{x} \in \mathcal{X}} \|\boldsymbol{W}\boldsymbol{x}\|_\infty^2$, then the optimal value of (17) is no less than the optimal value of (13). This follows immediately by noting that under the condition on M, the feasible set of (17) is contained in the feasible set of (13).*

The mixed integer second order cone reformulation and convex relaxation introduced in this section lead to two approaches for solving (4) to certifiable optimality. On the one hand, solvers like `Gurobi` contain direct support for solving mixed integer second order cone problems so problem (4) can be solved directly. On the other hand, it is possible to develop a custom branch-and-bound routine that leverages a modification of (7) to compute lower bounds. We illustrate this in Section 5. This custom,

13

problem specific approach outperforms `Gurobi` because (5) is a more tractable problem than (13) due in part to the presence of fewer second order cone constraints which decreases the computational time spent computing lower bounds.

4.2 A Positive Semidefinite Cone Relaxation

In this section, we formulate (4) as a polynomial optimization problem and present a semidefinite relaxation using the sum of squares (SOS) hierarchy [39]. We show that this semidefinite relaxation is tighter than the second order cone relaxation presented previously.

Let $f(\boldsymbol{z}, \boldsymbol{x}) = \sum_{i=1}^{n} z_i + \frac{1}{\gamma} \sum_{i=1}^{n} w_i^2 \theta_i$ denote the objective function of (12). Notice that the constraint $\boldsymbol{z} \in \{0,1\}^n$ in (12) is equivalent to the constraint $\boldsymbol{z} \circ \boldsymbol{z} = \boldsymbol{z}$ (where $\circ$ denotes the element wise product). With this observation, we can express the feasible set of (12) as the semialgebraic set given by:

$$\Omega = \{(\boldsymbol{z}, \boldsymbol{x}) \in \mathbb{R}^n \times \mathbb{R}^n : \epsilon - \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2 \geq 0, x_i z_i - x_i = 0 \ \forall \ i, z_i^2 - z_i = 0 \ \forall \ i\}.$$

Thus, we can equivalently write (12) as $\min_{(\boldsymbol{z}, \boldsymbol{x}) \in \Omega} f(\boldsymbol{z}, \boldsymbol{x})$. It is not difficult to see that the preceding optimization problem has the same optimal value as the problem given by

$$\max_{\lambda \in \mathbb{R}} \quad \lambda \text{ s.t. } f(\boldsymbol{z}, \boldsymbol{x}) - \lambda \geq 0 \ \forall \ (\boldsymbol{z}, \boldsymbol{x}) \in \Omega. \tag{18}$$

Problem (18) is a polynomial optimization problem that has the same optimal value as (4).

We can obtain tractable lower bounds for (18) by leveraging techniques from sum of squares (SOS) optimization [40, 41]. A polynomial $g \in \mathbb{R}[x]$ is said to be sum of squares (SOS) if for some $K \in \mathbb{N}$ there exists polynomials $\{g_k\}_{k=1}^{K} \subset \mathbb{R}[x]$ such that $g = \sum_{k=1}^{K} g_k^2$. We denote the set of all SOS polynomials as $\Sigma^2[x]$. Moreover, we denote the set of polynomials of degree at most d as $\mathbb{R}_d[x] \subset \mathbb{R}[x]$ and we denote the set of SOS polynomials of degree at most $2d$ as $\Sigma_d^2[x] \subset \Sigma^2[x]$. It is trivial to see that any polynomial that is SOS is globally non-negative. More generally, SOS polynomials can be utilized to model polynomial non-negativity over arbitrary semialgebraic sets. The quadratic module associated with the semialgebraic set Ω is defined as:

$$QM(\Omega) = \left\{ s_0(\boldsymbol{z}, \boldsymbol{x}) + s_1(\boldsymbol{z}, \boldsymbol{x})(\epsilon - \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2) + \sum_{i=1}^{n} t_i(\boldsymbol{z}, \boldsymbol{x})(x_i z_i - x_i) \right.$$
$$\left. + \sum_{i=1}^{n} r_i(\boldsymbol{z}, \boldsymbol{x})(z_i^2 - z_i) : s_0, s_1 \in \Sigma^2[\boldsymbol{z}, \boldsymbol{x}], t_i, r_i \in \mathbb{R}[\boldsymbol{z}, \boldsymbol{x}] \ \forall \ i \right\}. \tag{19}$$

It is straightforward to see that if a function $h(\boldsymbol{z}, \boldsymbol{x})$ is an element of $QM(\Omega)$, then $h(\boldsymbol{z}, \boldsymbol{x})$ is non-negative on Ω (since for points in Ω, $h(\boldsymbol{z}, \boldsymbol{x})$ takes the form of the sum of two SOS polynomials). Thus, membership in $QM(\Omega)$ is a sufficient condition for non-negativity on Ω. We further define the restriction of $QM(\Omega)$ to polynomials of

degree at most $2d$ as:

$$QM_d(\Omega) = \left\{ s_0(\boldsymbol{z}, \boldsymbol{x}) + s_1(\boldsymbol{z}, \boldsymbol{x})(\epsilon - \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2) + \sum_{i=1}^n t_i(\boldsymbol{z}, \boldsymbol{x})(x_i z_i - x_i) \right.$$

$$\left. + \sum_{i=1}^n r_i(\boldsymbol{z}, \boldsymbol{x})(z_i^2 - z_i) : s_0 \in \Sigma_d^2[\boldsymbol{z}, \boldsymbol{x}], s_1 \in \Sigma_{d-1}^2[\boldsymbol{z}, \boldsymbol{x}], t_i, r_i \in \mathbb{R}_{2d-2}[\boldsymbol{z}, \boldsymbol{x}] \ \forall \ i \right\}. \tag{20}$$

It is immediate that $QM_d(\Omega) \subset QM(\Omega)$ and membership in $QM_d(\Omega)$ provides a certificate of non-negativity on Ω. Importantly, given an arbitrary polynomial $h(\boldsymbol{z}, \boldsymbol{x})$ it is possible to verify membership in $QM_d(\Omega)$ by checking feasibility of a semidefinite program. Thus, for any $d \in \mathbb{N}$, we obtain a semidefinite relaxation of (4) by solving:

$$\max_{\lambda \in \mathbb{R}} \quad \lambda \text{ s.t. } f(\boldsymbol{z}, \boldsymbol{x}) - \lambda \in QM_d(\Omega). \tag{21}$$

Since $QM_d(\Omega) \subset QM_{d+1}(\Omega)$, (21) produces an increasingly strong lower bound with increasing values of d. A natural question to ask is how the relaxation given by (21) compares to that given by (13). We answer this question in Theorem 5.

Theorem 5. *For every $d \geq 1$, the optimal value of (21) is no less than the optimal value of (13).*

Proof. Without loss of generality, we take $\boldsymbol{W} = \boldsymbol{I}$. We prove the result for $\gamma \geq \gamma_0 = \max_{\boldsymbol{x} \in \mathcal{X}} \|\boldsymbol{x}\|_\infty^2$ though the result extends naturally to the case of arbitrary γ. Fix any $\epsilon > 0$, $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ and $\boldsymbol{b} \in \mathbb{R}^m$. By Theorem 4, (13) has the same optimal value as (7). Consider the dual of (7) which for $\boldsymbol{W} = \boldsymbol{I}$ is given by

$$\max_{\boldsymbol{\nu} \in \mathbb{R}^m} \quad \boldsymbol{b}^T \boldsymbol{\nu} - \sqrt{\epsilon} \|\boldsymbol{\nu}\|_2 \text{ s.t. } |\boldsymbol{\nu}^T A_i| \leq \frac{2}{\sqrt{\gamma}} \quad \forall \ i, \tag{22}$$

where A_i denotes the ith column of $\boldsymbol{A}$. Strong duality holds between (22) and (7) since $\boldsymbol{\nu} = 0$ is always a strictly feasible point in (22) [42]. Fix $d = 1$. We will show that for any feasible solution to (22), we can construct a feasible solution to (21) that achieves the same objective value. Let $\bar{\boldsymbol{\nu}} \in \mathbb{R}^m$ denote an arbitrary feasible solution to (22). Define $\bar{r}_i(\boldsymbol{z}, \boldsymbol{x}) = -1, \bar{t}_i(\boldsymbol{z}, \boldsymbol{x}) = A_i^T \bar{\boldsymbol{\nu}}$ for all i, $\bar{s}_1(\boldsymbol{z}, \boldsymbol{x}) = \frac{\|\bar{\boldsymbol{\nu}}\|_2}{2\sqrt{\epsilon}}$ and define $\bar{s}_0(\boldsymbol{z}, \boldsymbol{x}) = \text{monomial}(\boldsymbol{z}, \boldsymbol{x}, 1)^T \bar{\boldsymbol{S}} \text{monomial}(\boldsymbol{z}, \boldsymbol{x}, 1)$ where $\text{monomial}(\boldsymbol{x}, \boldsymbol{z}, 1) \in \mathbb{R}[\boldsymbol{z}, \boldsymbol{x}]^{2n+1}$ is the vector of monomials in $\mathbb{R}[\boldsymbol{z}, \boldsymbol{x}]$ of degree at most 1 and $\bar{\boldsymbol{S}} \in \mathbb{R}^{2n+1 \times 2n+1}$ is given by

$$\bar{\boldsymbol{S}} = \left[\begin{array}{c|c|c} \frac{1}{\gamma} \boldsymbol{I}_n + \frac{\|\bar{\boldsymbol{\nu}}\|_2}{2\sqrt{\epsilon}} \boldsymbol{A}^T \boldsymbol{A} & \text{diag}\left(\frac{-\boldsymbol{A}^T \bar{\boldsymbol{\nu}}}{2}\right) & \frac{1}{2} \boldsymbol{A}^T \left(\bar{\boldsymbol{\nu}} - \frac{\|\bar{\boldsymbol{\nu}}\|_2}{\sqrt{\epsilon}} \boldsymbol{b}\right) \\ \hline \text{diag}\left(\frac{-\boldsymbol{A}^T \bar{\boldsymbol{\nu}}}{2}\right) & \boldsymbol{I}_n & \boldsymbol{0}_n \\ \hline \frac{1}{2}\left(\bar{\boldsymbol{\nu}}^T - \frac{\|\bar{\boldsymbol{\nu}}\|_2}{\sqrt{\epsilon}} \boldsymbol{b}^T\right) \boldsymbol{A} & \boldsymbol{0}_n^T & \left(\frac{\boldsymbol{b}^T \boldsymbol{b}}{2\sqrt{\epsilon}} + \frac{\sqrt{\epsilon}}{2}\right) \|\bar{\boldsymbol{\nu}}\|_2 - \bar{\boldsymbol{\nu}}^T \boldsymbol{b} \end{array} \right].$$

Clearly, we have $\bar{t}_i, \bar{r}_i \in \mathbb{R}_0[\boldsymbol{z}, \boldsymbol{x}]$ for all i and $\bar{s}_1 \in \Sigma_0^2[\boldsymbol{z}, \boldsymbol{x}]$ because $\frac{\|\bar{\boldsymbol{\nu}}\|_2}{2\sqrt{\epsilon}} \geq 0$. We claim that $\bar{s}_0 \in \Sigma_1^2[\boldsymbol{z}, \boldsymbol{x}]$. To see this, note that by the generalized Schur complement

15

lemma (see Boyd et al. 1994, Equation 2.41), $\bar{\boldsymbol{S}} \succeq 0$ if and only if $\begin{pmatrix} \boldsymbol{I}_n & \boldsymbol{0}_n \\ \boldsymbol{0}_n^T & \sigma \end{pmatrix} \succeq 0$ and $\frac{1}{\gamma} \boldsymbol{I}_n + \frac{\|\bar{\boldsymbol{\nu}}\|_2}{2\sqrt{\epsilon}} \boldsymbol{A}^T \boldsymbol{A} - \text{diag}\left(\frac{-\boldsymbol{A}^T \bar{\boldsymbol{\nu}}}{2}\right)^2 - \frac{1}{4\sigma} \boldsymbol{A}^T \left(\bar{\boldsymbol{\nu}} - \frac{\|\bar{\boldsymbol{\nu}}\|_2}{\sqrt{\epsilon}} \boldsymbol{b}\right) \left(\bar{\boldsymbol{\nu}} - \frac{\|\bar{\boldsymbol{\nu}}\|_2}{\sqrt{\epsilon}} \boldsymbol{b}\right)^T \boldsymbol{A} \succeq 0$ where we let $\sigma = \left(\frac{\boldsymbol{b}^T \boldsymbol{b}}{2\sqrt{\epsilon}} + \frac{\sqrt{\epsilon}}{2}\right) \|\bar{\boldsymbol{\nu}}\|_2 - \bar{\boldsymbol{\nu}}^T \boldsymbol{b}$. The first condition is satisfied if $\sigma \geq 0$. To see that this is always the case, notice that we can equivalently express σ as

$$\sigma = \left(\frac{\boldsymbol{b}^T \boldsymbol{b} + \epsilon}{2\sqrt{\epsilon}\|\boldsymbol{b}\|_2}\right) \|\boldsymbol{b}\|_2 \|\bar{\boldsymbol{\nu}}\|_2 - \bar{\boldsymbol{\nu}}^T \boldsymbol{b}.$$

By Cauchy-Schwarz, we have $|\bar{\boldsymbol{\nu}}^T \boldsymbol{b}| \leq \|\boldsymbol{b}\|_2 \|\bar{\boldsymbol{\nu}}\|_2$. Moreover, we have $0 \leq (\|\boldsymbol{b}\|_2 - \sqrt{\epsilon})^2 \implies \frac{\boldsymbol{b}^T \boldsymbol{b} + \epsilon}{2\sqrt{\epsilon}\|\boldsymbol{b}\|_2} \geq 1$. It follows immediately that $\sigma \geq 0$.

To establish the second condition, we first rewrite the Schur complement of $\bar{\boldsymbol{S}}$ as the sum of two matrices:

$$\left[\frac{1}{\gamma} \boldsymbol{I}_n - \text{diag}\left(\frac{-\boldsymbol{A}^T \bar{\boldsymbol{\nu}}}{2}\right)^2\right] + \left[\frac{\|\bar{\boldsymbol{\nu}}\|_2}{2\sqrt{\epsilon}} \boldsymbol{A}^T \boldsymbol{A} - \frac{1}{4\sigma} \boldsymbol{A}^T \left(\bar{\boldsymbol{\nu}} - \frac{\|\bar{\boldsymbol{\nu}}\|_2}{\sqrt{\epsilon}} \boldsymbol{b}\right) \left(\bar{\boldsymbol{\nu}} - \frac{\|\bar{\boldsymbol{\nu}}\|_2}{\sqrt{\epsilon}} \boldsymbol{b}\right)^T \boldsymbol{A}\right],$$

where we let $\boldsymbol{\Phi}$ denote the first matrix in the sum and we let $\boldsymbol{\Psi}$ denote the second matrix. It suffices to show that $\boldsymbol{\Phi}$ and $\boldsymbol{\Psi}$ are both positive semidefinite. The eigenvalues of $\boldsymbol{\Phi}$ are given by $\{\frac{1}{\gamma} - \frac{(\bar{\boldsymbol{\nu}}^T A_i)^2}{4}\}_{i=1}^n$. Thus, $\boldsymbol{\Phi}$ is positive semidefinite as long as $|\bar{\boldsymbol{\nu}}^T A_i| \leq \frac{2}{\sqrt{\gamma}}$ for all i which is guaranteed by the feasibility of $\bar{\boldsymbol{\nu}}$ in (22). To see that $\boldsymbol{\Psi} \succeq 0$, note that we can write $\boldsymbol{\Psi}$ as

$$\boldsymbol{\Psi} = \boldsymbol{A}^T \left(\frac{\|\bar{\boldsymbol{\nu}}\|_2}{2\sqrt{\epsilon}} \boldsymbol{I}_m - \frac{1}{4\sigma}\left(\bar{\boldsymbol{\nu}} - \frac{\|\bar{\boldsymbol{\nu}}\|_2}{\sqrt{\epsilon}} \boldsymbol{b}\right) \left(\bar{\boldsymbol{\nu}} - \frac{\|\bar{\boldsymbol{\nu}}\|_2}{\sqrt{\epsilon}} \boldsymbol{b}\right)^T\right) \boldsymbol{A}.$$

Since any matrix of the form $\boldsymbol{X}^T \boldsymbol{Y} \boldsymbol{X}$ is positive semidefinite provided that $\boldsymbol{Y}$ is positive semidefinite, it suffices to show that $\frac{\|\bar{\boldsymbol{\nu}}\|_2}{2\sqrt{\epsilon}} \boldsymbol{I}_m \succeq \frac{1}{4\sigma}\left(\bar{\boldsymbol{\nu}} - \frac{\|\bar{\boldsymbol{\nu}}\|_2}{\sqrt{\epsilon}} \boldsymbol{b}\right) \left(\bar{\boldsymbol{\nu}} - \frac{\|\bar{\boldsymbol{\nu}}\|_2}{\sqrt{\epsilon}} \boldsymbol{b}\right)^T$. Notice that the left hand side term of the matrix inequality has m eigenvalues of the form $\frac{\|\bar{\boldsymbol{\nu}}\|_2}{2\sqrt{\epsilon}}$ while the right hand side term of the inequality is a rank 1 matrix with $\frac{1}{4\sigma}\left(\bar{\boldsymbol{\nu}} - \frac{\|\bar{\boldsymbol{\nu}}\|_2}{\sqrt{\epsilon}} \boldsymbol{b}\right)^T \left(\bar{\boldsymbol{\nu}} - \frac{\|\bar{\boldsymbol{\nu}}\|_2}{\sqrt{\epsilon}} \boldsymbol{b}\right)$ as its only non-zero eigenvalue. Recalling the definition of σ, it can be easily verified that $\left(\bar{\boldsymbol{\nu}} - \frac{\|\bar{\boldsymbol{\nu}}\|_2}{\sqrt{\epsilon}} \boldsymbol{b}\right)^T \left(\bar{\boldsymbol{\nu}} - \frac{\|\bar{\boldsymbol{\nu}}\|_2}{\sqrt{\epsilon}} \boldsymbol{b}\right) = \frac{2\sigma\|\bar{\boldsymbol{\nu}}\|_2}{\sqrt{\epsilon}}$ by simple algebraic manipulation. Accordingly, we have $\frac{\|\bar{\boldsymbol{\nu}}\|_2}{2\sqrt{\epsilon}} \boldsymbol{I}_m \succeq \frac{1}{4\sigma}\left(\bar{\boldsymbol{\nu}} - \frac{\|\bar{\boldsymbol{\nu}}\|_2}{\sqrt{\epsilon}} \boldsymbol{b}\right) \left(\bar{\boldsymbol{\nu}} - \frac{\|\bar{\boldsymbol{\nu}}\|_2}{\sqrt{\epsilon}} \boldsymbol{b}\right)^T \implies \boldsymbol{\Psi} \succeq 0$. Thus, we have established that $\bar{\boldsymbol{S}} \succeq 0 \implies \bar{s}_0 \in \Sigma_1^2[\boldsymbol{z}, \boldsymbol{x}]$. Finally, we note that

$$\bar{s}_0 + \bar{s}_1(\epsilon - \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2) + \sum_{i=1}^n \bar{t}_i(x_i z_i - x_i) + \sum_{i=1}^n \bar{r}_i(z_i^2 - z_i) = f(\boldsymbol{z}, \boldsymbol{x}) - \boldsymbol{b}^T \bar{\boldsymbol{\nu}} + \sqrt{\epsilon}\|\bar{\boldsymbol{\nu}}\|_2.$$

We have shown that given an arbitrary feasible solution to (22), we can construct a solution that is feasible to (21) that achieves the same objective value. Note that this

16

construction holds for any $d \geq 1$. Thus, for any $d \in \mathbb{N}$ the optimal value of (21) is at least as high as the optimal value of (13). $\qquad\qquad\square$

We have shown that for any value of d, (21) produces a lower bound on the optimal value of (4) at least as strong as the bound given by (13). Unfortunately, (21) suffers from scalability challenges as it requires solving a positive semidefinite program with PSD constraints on matrices with dimension $\binom{2n+d}{d} \times \binom{2n+d}{d}$. We further discuss the scalability of (21) in Section 6. Note that since (21) is a maximization problem, any feasible solution (in particular a nearly optimal one) still consists of a valid lower bound on the optimal value of (4).

5 Branch-and-Bound

In this section, we propose a branch-and-bound algorithm in the sense of [43, 44] that computes certifiably optimal solutions to Problem (3) by solving the mixed integer second order cone reformulation given by (12). We state explicitly our subproblem strategy in Section 5.1, before stating our overall algorithmic approach in Section 5.2.

5.1 Subproblems

Henceforth, for simplicity we will assume the weights w_i take value 1 for all i. What follows generalizes immediately to the setting where this assumption does not hold. Notice that (12) can be equivalently written as the two stage optimization problem given by $\min_{\boldsymbol{z} \in \{0,1\}^n} h(\boldsymbol{z})$ where we define $h(\boldsymbol{z})$ as:

$$
\begin{aligned}
h(\boldsymbol{z}) = \min_{\boldsymbol{x}, \boldsymbol{\theta} \in \mathbb{R}^n} \quad & \sum_{i=1}^n z_i + \frac{1}{\gamma} \sum_{i=1}^n \theta_i \\
\text{s.t.} \quad & \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2 \leq \epsilon, \ x_i^2 \leq z_i \theta_i \ \forall \ i, \ \theta_i \geq 0 \ \forall i.
\end{aligned}
\tag{23}
$$

Note that in general, there exist binary vectors $\bar{\boldsymbol{z}} \in \{0,1\}^n$ such that the optimization problem in (23) is infeasible. For any such $\bar{\boldsymbol{z}}$, we define $h(\bar{\boldsymbol{z}}) = \infty$. We construct an enumeration tree that branches on the entries of the binary vector $\boldsymbol{z}$ which models the support of $\boldsymbol{x}$. A (partial or complete) sparsity pattern is associated with each node in the tree and is defined by disjoint collections $\mathcal{I}_0, \mathcal{I}_1 \subseteq [n]$. For indices $i \in \mathcal{I}_0$, we constrain $z_i = 0$ and for indices $j \in \mathcal{I}_1$, we constrain $z_j = 1$. We say that $\mathcal{I}_0$ and $\mathcal{I}_1$ define a complete sparsity pattern if $|\mathcal{I}_0| + |\mathcal{I}_1| = n$, otherwise we say that $\mathcal{I}_0$ and $\mathcal{I}_1$ define a partial sparsity pattern. A node in the tree is said to be terminal if its associated sparsity pattern is complete.

Each node in the enumeration tree has an associated subproblem, defined by the collections $\mathcal{I}_0$ and $\mathcal{I}_1$, which is given by:

$$
\min_{\boldsymbol{z} \in \{0,1\}^n} \quad h(\boldsymbol{z}), \quad \text{s.t.} \quad z_i = 0 \ \forall \, i \in \mathcal{I}_0, z_j = 1 \ \forall \, j \in \mathcal{I}_1.
\tag{24}
$$

Note that if $\mathcal{I}_0 = \mathcal{I}_1 = \emptyset$, (24) is equivalent to (12) (under the assumetion that $w_i = 1$ for all i).

17

5.1.1 Subproblem Lower Bound

Let $\mathcal{I} = \mathcal{I}_0 \cup \mathcal{I}_1$. We obtain a lower bound for (24) by relaxing the binary variables that are not fixed (z_i such that $i \notin \mathcal{I}$) to take values within the interval $[0,1]$. The resulting lower bound is given by

$$
\begin{aligned}
\min_{\boldsymbol{z}, \boldsymbol{x}, \boldsymbol{\theta} \in \mathbb{R}^n} \quad & \sum_{i=1}^{n} z_i + \frac{1}{\gamma} \sum_{i=1}^{n} \theta_i \\
\text{s.t.} \quad & \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2 \leq \epsilon, \; x_i^2 \leq z_i \theta_i \; \forall \, i, \; 0 \leq z_i \leq 1 \; \forall \, i \notin \mathcal{I}, \\
& z_i = 0 \; \forall \, i \in \mathcal{I}_0, \; z_i = 1 \; \forall \, i \in \mathcal{I}_1, \; \theta_i \geq 0 \; \forall \, i.
\end{aligned}
\tag{25}
$$

Notice that for an arbitrary set $\bar{\mathcal{I}}_0 \subseteq [n]$, problems (24) and (25) are infeasible if and only if the set $\{\boldsymbol{x} : \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2 \leq \epsilon, x_i = 0 \; \forall \, i \in \bar{\mathcal{I}}_0\}$ is empty. Moreover, observe it immediately follows that if (24) and (25) are infeasible for $\bar{\mathcal{I}}_0$, then they are also infeasible for any set $\hat{\mathcal{I}}_0 \subseteq [n]$ satisfying $\bar{\mathcal{I}}_0 \subseteq \hat{\mathcal{I}}_0$. We use this observation in section 5.2 to generate feasibility cuts whenever an infeasible subproblem is encountered in the branch-and-bound tree. Using a similar argument as in the proof of Theorem 4, it can be shown that when $\gamma \geq \gamma_0 = \max_{x \in \mathcal{X}} \|\boldsymbol{x}\|_\infty^2$, (25) is equivalent to the convex optimization problem given by (26):

$$
\begin{aligned}
\min_{\boldsymbol{x} \in \mathbb{R}^n} \quad & |\mathcal{I}_1| + \frac{1}{\gamma} \sum_{i \in \mathcal{I}_1} x_i^2 + \frac{2}{\sqrt{\gamma}} \sum_{i \notin \mathcal{I}} |x_i| \\
\text{s.t.} \quad & \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2 \leq \epsilon, \; x_i = 0 \; \forall \, i \in \mathcal{I}_0,
\end{aligned}
\tag{26}
$$

where if $\boldsymbol{x}^\star$ is optimal to (26), then $(\boldsymbol{z}^\star, \boldsymbol{x}^\star, \boldsymbol{\theta}^\star)$ is optimal to (25) taking $z_i^\star = \frac{|x_i|^\star}{\sqrt{\gamma}}$ and $\theta_i^\star = x_i^{\star 2}$. Problem (26) is a second order cone problem that emits the following dual:

$$
\max_{\boldsymbol{\nu} \in \mathbb{R}^m} \quad |\mathcal{I}_1| + \boldsymbol{b}^T \boldsymbol{\nu} - \sqrt{\epsilon} \|\boldsymbol{\nu}\|_2 - \frac{\gamma}{4} \boldsymbol{\nu}^T \sum_{i \in \mathcal{I}_1} (A_i A_i^T) \boldsymbol{\nu} \;\; \text{s.t.} \;\; |\boldsymbol{\nu}^T A_i| \leq \frac{2}{\sqrt{\gamma}} \; \forall \, i \notin \mathcal{I}.
\tag{27}
$$

Strong duality holds between (26) and (27) since $\boldsymbol{\nu} = 0$ is always a strictly feasible point in (27) for any collections $\mathcal{I}_0, \mathcal{I}_1$ [42]. In our branch-and-bound implementation described in 5.2, we compute lower bounds by solving (26) using `Gurobi`. We note that depending on the solver employed, it may be beneficial to compute lower bounds using (27) in place of (26).

5.1.2 Subproblem Upper Bound

Recall that solving Problem (2) can be interpreted as determining the minimum number of columns from the input matrix $\boldsymbol{A}$ that must be selected such that the residual of the projection of the input vector $\boldsymbol{b}$ onto the span of the selected columns has ℓ_2 norm equal to at most $\sqrt{\epsilon}$. The same interpretation holds for Problem (3) under the assumption that the ℓ_2 regularization term in the objective is negligible.

Consider an arbitrary node in the branch-and-bound algorithm and let $\boldsymbol{x}^\star$ denote an optimal solution to (26). To obtain an upper bound to (24), we define an ordering

on the columns of $\boldsymbol{A}$ and iteratively select columns from this ordering from largest to smallest until the ℓ_2 norm of the residual of the projection of $\boldsymbol{b}$ onto the selected columns is less than $\sqrt{\epsilon}$. The ordering of the columns of $\boldsymbol{A}$ corresponds to sorting the entries of $\boldsymbol{x}^\star$ in decreasing absolute value. Specifically, we have $A_i \succeq A_j \iff |x_i^\star| \geq |x_j^\star|$. Algorithm 1 outlines this approach. For an arbitrary collection of indices $\mathcal{I}_t \subseteq [n]$, we let $\boldsymbol{A}(\mathcal{I}_t) \in \mathbb{R}^{m \times |\mathcal{I}_t|}$ denote the matrix obtained by stacking the $|\mathcal{I}_t|$ columns of $\boldsymbol{A}$ corresponding to the indices in the set $\mathcal{I}_t$. Specifically, if i_k denotes the kth entry of $\mathcal{I}_t$, then the kth column of $\boldsymbol{A}(\mathcal{I}_t)$ is A_i. Let $\boldsymbol{x}^{ub}$ denote the output of Algorithm 1. The objective value achieved by $\boldsymbol{x}^{ub}$ in (3) is the upper bound.

Algorithm 1 Branch-and-Bound Upper Bound

Require: $\boldsymbol{A} \in \mathbb{R}^{m \times n}, \boldsymbol{b} \in \mathbb{R}^m, \epsilon > 0$. An optimal solution $\boldsymbol{x}^\star$ of (26).
Ensure: $\bar{\boldsymbol{x}}$ is feasible to (3).
1: $\mathcal{I}_0 \leftarrow \emptyset$;
2: $\boldsymbol{r}_0 \leftarrow \boldsymbol{b}$;
3: $t \leftarrow 0$;
4: $\delta_0 \leftarrow \|\boldsymbol{r}_0\|_2^2$
5: **while** $\delta_t > \epsilon$ **do**
6: $\quad i_t \leftarrow \arg\max_{i \in [n] \setminus \mathcal{I}_t} |x_i^\star|$;
7: $\quad \mathcal{I}_{t+1} \leftarrow \mathcal{I}_t \cup i_t$;
8: $\quad \boldsymbol{x}_{t+1} \leftarrow \left[\boldsymbol{A}(\mathcal{I}_{t+1})^T \boldsymbol{A}(\mathcal{I}_{t+1})\right]^\dagger \boldsymbol{A}(\mathcal{I}_{t+1})^T \boldsymbol{b}$;
9: $\quad \boldsymbol{r}_{t+1} \leftarrow \boldsymbol{b} - \boldsymbol{A}(\mathcal{I}_{t+1})\boldsymbol{x}_{t+1}$;
10: $\quad \delta_{t+1} \leftarrow \|\boldsymbol{r}_{t+1}\|_2^2$;
11: $\quad t \leftarrow t + 1$;
12: **end while**
13: Define $\bar{\boldsymbol{x}} \in \mathbb{R}^n$ as $\bar{x}(i_k) = x_t(k)$ for $i_k \in \mathcal{I}_t$ and $\bar{x}(i_k) = 0$ otherwise;
14: Return $\bar{\boldsymbol{x}}$.

The computational bottleneck of Algorithm 1 is computing the matrix inverse of $\boldsymbol{A}(\mathcal{I}_t)^T \boldsymbol{A}(\mathcal{I}_t) \in \mathbb{R}^{|\mathcal{I}_t| \times |\mathcal{I}_t|}$ at each iteration. Doing so explicitly at each iteration t would require $O(|\mathcal{I}_t|^3)$ operations. Letting $k^\star = \|\boldsymbol{x}^{ub}\|_0$ where $\boldsymbol{x}^{ub}$ is the output of Algorithm 1, the total cost of executing these matrix inversions is

$$\sum_{t=1}^{k^\star} |\mathcal{I}_t|^3 = \sum_{t=1}^{k^\star} t^3 = \left[\frac{k^\star(k^\star + 1)}{2}\right]^2 = O(k^{\star 4})$$

However, it is possible to accelerate the computation of these matrix inverses by leveraging the fact that $\boldsymbol{A}(\mathcal{I}_t)$ and $\boldsymbol{A}(\mathcal{I}_{t+1})$ differ only by the addition of one column and leveraging block matrix inversion which states that for matrices $\boldsymbol{C} \in \mathbb{R}^{n_1 \times n_1}, \boldsymbol{D} \in \mathbb{R}^{n_2 \times n_2}$ and $\boldsymbol{U}, \boldsymbol{V} \in \mathbb{R}^{n_1 \times n_2}$, we have:

$$\begin{bmatrix} \boldsymbol{C} & \boldsymbol{U} \\ \boldsymbol{V}^T & \boldsymbol{D} \end{bmatrix}^\dagger = \begin{bmatrix} \boldsymbol{C}^\dagger + \boldsymbol{C}^\dagger \boldsymbol{U}(\boldsymbol{D} - \boldsymbol{V}^T \boldsymbol{C}^\dagger \boldsymbol{U})^{-1} \boldsymbol{V}^T \boldsymbol{C}^\dagger & -\boldsymbol{C}^\dagger \boldsymbol{U}(\boldsymbol{D} - \boldsymbol{V}^T \boldsymbol{C}^\dagger \boldsymbol{U})^{-1} \\ -(\boldsymbol{D} - \boldsymbol{V}^T \boldsymbol{C}^\dagger \boldsymbol{U})^{-1} \boldsymbol{V}^T \boldsymbol{C}^\dagger & (\boldsymbol{D} - \boldsymbol{V}^T \boldsymbol{C}^\dagger \boldsymbol{U})^{-1} \end{bmatrix}$$

where it is assumed that the matrix $(\boldsymbol{D} - \boldsymbol{V}^T \boldsymbol{C}^\dagger \boldsymbol{U})$ is invertible [45]. Letting $n_1 = |\mathcal{I}_t|, n_2 = 1, \boldsymbol{C} = \boldsymbol{A}(\mathcal{I}_t)^T \boldsymbol{A}(\mathcal{I}_t), \boldsymbol{U} = \boldsymbol{V} = \boldsymbol{A}(\mathcal{I}_t)^T a_{i_t}$, and $\boldsymbol{D} = a_{i_t}^T a_{i_t}$, we can compute the matrix inverse of $\boldsymbol{A}(\mathcal{I}_{t+1})^T \boldsymbol{A}(\mathcal{I}_{t+1})$ using $O(|\mathcal{I}_t|^2 + m|\mathcal{I}_t|)$ operations. With this implementation, the total cost of executing matrix inversions in Algorithm 1 becomes

$$\sum_{t=1}^{k^\star} |\mathcal{I}_t|^2 + m|\mathcal{I}_t| = \sum_{t=1}^{k^\star} t^2 + m \sum_{t=1}^{k^\star} t = O(k^{\star 3} + mk^{\star 2})$$

which is a significant improvement over the naive $O(k^{\star 4})$ approach.

5.2 Branch-and-Bound Algorithm

Having stated how we can compute upper and lower bounds to (23) at each node in the enumeration tree, we are now ready to present the branch-and-bound algorithm in its entirety. Algorithm 2 describes our approach which is based on the implementation by [46]. Though branching rules and node selection rules for branch-and-bound algorithms form a rich literature [47], we follow the design of [46] and employ the most fractional branching rule and least lower bound node selection rule.

Explicitly, for an arbitrary non-terminal node p, let $\boldsymbol{z}^*$ be the optimal vector $\boldsymbol{z}$ of the node relaxation given by (25). We branch on entry $i^\star = \arg\min_{i \notin \mathcal{I}_0 \cup \mathcal{I}_1} |z_i - 0.5|$. When selecting a node to investigate, we select the node whose lower bound is equal to the global lower bound. If multiple such nodes exist, we choose arbitrarily from the collection of nodes satisfying this condition. Suppose that a given node produces a subproblem (26) that is infeasible where we let $\bar{\mathcal{I}}_0$ correspond to the zero index set of this node. Note that this implies that all child nodes of this node will also produce infeasible subproblems. Accordingly, to prune this region of the parameter space entirely, we introduce the feasibility cut $\sum_{i \in \bar{\mathcal{I}}_0} z_i \geq 1$. Let $f(\boldsymbol{x}) = \|\boldsymbol{x}\|_0 + \frac{1}{\gamma}\|\boldsymbol{x}\|_2^2$, the objective function of (3) and let $g(\mathcal{I}_0, \mathcal{I}_1)$ denote the optimal value of (26) for any collections $\mathcal{I}_0, \mathcal{I}_1 \subseteq [n], \mathcal{I}_0 \cap \mathcal{I}_1 = \emptyset$. The final objective value returned by Algorithm 2 is given by $\min_i f(\boldsymbol{x}_i)$ where $\{\boldsymbol{x}_i\}_i$ denotes the collection of feasible solutions produced by Algorithm 1 at any point during the execution of Algorithm 2. The output lower bound of Algorithm 2 is given by $\min_{(\mathcal{I}_0, \mathcal{I}_1) \in \mathcal{N}} g(\mathcal{I}_0, \mathcal{I}_1)$ where $\mathcal{N}$ denotes the set of non-discarded nodes upon the termination of Algorithm 2.

Let lb denote a lower bound to a given arbitrary minimization problem and let ub denote the objective value achieved by a feasible solution $\bar{\boldsymbol{x}}$ to the minimization problem. We call the solution $\bar{\boldsymbol{x}}$ a δ *globally optimal solution* to the given minimization problem if we have $lb \leq ub \leq (1 + \delta)lb$.

Theorem 6. *Algorithm 2 terminates in a finite number of iterations and returns a δ globally optimal solution to (2).*

Proof. The proof follows the proof of Theorem 21 in [46]. Note that Algorithm 2 can never visit a node more than once and that there is a finite number of partial and complete sparsity patterns (each corresponding to a possible node in the tree) because the set $\{0, 1\}^n$ is discrete. Thus, Algorithm 2 terminates in a finite number of iterations. Moreover, upon termination we must have $\frac{ub - lb}{ub} \leq \delta$, therefore the output

solution $\bar{\boldsymbol{x}}$ is δ globally optimal to problem (3) by definition since lb consists of a global lower bound and $\bar{\boldsymbol{x}}$ is feasible to (3). □

We conclude the discussion of Algorithm 2 by describing two modifications that accelerate its execution time (or equivalently, improve its scalability) at the expense of sacrificing the universal optimality guarantee by drawing on techniques from the high dimensional sparse machine learning literature [48] and the deep learning literature [49].

5.2.1 Backbone Optimization

Note that the total number of terminal nodes in the branch-and-bound tree is at most $\sum_{k=1}^{n} \binom{n}{k} = 2^n$ in the worst case so the total number of nodes can be upper bounded by $2^{n+1} - 1$. Since the runtime of Algorithm 2 (and the feasible space) is proportional to the number of nodes explored which grows exponentially in n, reducing n leads to reduced run time. Observe that if we knew in advance that the support of the optimal solution to (3) was contained within a set of cardinality less than n, then we could run Algorithm (2) on the corresponding reduced feature set which would result in improving the runtime of (2) while preserving its optimality guarantee. Formally, let $\boldsymbol{x}^\star$ denote an optimal solution to (3). If we know a priori that support$(\boldsymbol{x}^\star) \subseteq \mathcal{I} \subset [n]$, then we can pass $\boldsymbol{A}(\mathcal{I})$ to Algorithm 2 in place of $\boldsymbol{A}$ without discarding $\boldsymbol{x}^\star$ from the feasible set. The speed up can be quite significant when $|\mathcal{I}| \ll n$.

Knowing with certainty that support$(\boldsymbol{x}^\star) \subseteq \mathcal{I} \subset [n]$ a priori is too strong an assumption, however a more reasonable assumption is knowing a priori that with high probability there exists a good solution $\bar{\boldsymbol{x}}$ with support$(\bar{\boldsymbol{x}}) \subseteq \mathcal{I} \subset [n]$. In this setting, we can still pass $\boldsymbol{A}(\mathcal{I})$ to Algorithm 2 and benefit from an improved runtime at the expense of sacrificing optimality guarantees. In this setting, the columns of $\boldsymbol{A}(\mathcal{I})$ can be interpreted as a backbone for (3) [48]. In practice, $\mathcal{I}$ can be taken to be the set of features selected by some heuristic method. In Section 6, we take $\mathcal{I} = \{i : |\bar{x}_i| \geq 10^{-6}\}$ where $\bar{\boldsymbol{x}}$ is an optimal solution to (5).

5.2.2 Early Stopping

A common property of branch-and-bound algorithms is that the algorithm quickly arrives at an optimal (or near-optimal) solution early during the optimization procedure and spends the majority of its execution time improving the lower bound to obtain a certificate of optimality. Accordingly, this motivates halting Algorithm 2 before it terminates and taking its upper bound at the time of termination to be its output. Doing so is likely to still yield a high quality solution while reducing the Algorithm's runtime. In Section 6, we place an explicit time limit on Algorithm 2 and return the current upper bound if the Algorithm has not already terminated before reaching the time limit. Note that this approach shares strong connections with early stopping in the training of neural networks [49]. A well studied property of over-parameterized neural networks is that as the optimization procedure progresses, the error on the training data continues to decrease though the validation error plateaus and sometimes even increases. Given that the validation error is the metric of greater import, a common network training technique is to stop the optimization procedure after the

21

Algorithm 2 Optimal Compressed Sensing

Require: $\boldsymbol{A} \in \mathbb{R}^{m \times n}, \boldsymbol{b} \in \mathbb{R}^m, \epsilon, \gamma \in \mathbb{R}^+$. Tolerance parameter $\delta \geq 0$.
Ensure: $\bar{\boldsymbol{x}}$ solves (3) within the optimality tolerance δ.

1: **if** $\|(\boldsymbol{I} - \boldsymbol{A}[\boldsymbol{A}^T\boldsymbol{A}]^\dagger \boldsymbol{A}^T)\boldsymbol{b}\|_2^2 > \epsilon$ **then**
2:     Return $\emptyset$;
3: **end if**
4: **if** $\|\boldsymbol{b}\|_2^2 \leq \epsilon$ **then**
5:     Return $\boldsymbol{0}$;
6: **end if**
7: $p_0 \leftarrow (\mathcal{I}_0, \mathcal{I}_1) = (\emptyset, \emptyset)$;
8: $\mathcal{N} \leftarrow \{p_0\}$;
9: $lb \leftarrow$ optimal value of (26);
10: $\bar{\boldsymbol{x}} \leftarrow$ solution returned by Algorithm 1;
11: $ub \leftarrow f(\bar{\boldsymbol{x}})$;
12: **while** $\frac{ub-lb}{ub} > \epsilon$ **do**
13:     select $(\mathcal{I}_0, \mathcal{I}_1) \in \mathcal{N}$ according to the node selection rule;
14:     select an index $i \notin \mathcal{I}_0 \cup \mathcal{I}_1$ according to the branching rule;
15:     **for** $k = 0, 1$ **do**
16:         $l \leftarrow (k+1) \bmod 2$;
17:         newnode $\leftarrow \left((\mathcal{I}_k \cup i), \mathcal{I}_l\right)$;
18:         **if** newnode violates an existing feasibility cut **then**
19:           continue;
20:         **end if**
21:         **if** newnode is infeasible **then**
22:           Add the feasibility cut $\sum_{i \in \mathcal{I}_0} z_i \geq 1$;
23:         **end if**
24:         $lower \leftarrow$ lowerBound(newnode);
25:         $upper \leftarrow$ upperBound(newnode) with feasible point $\boldsymbol{x}^\star$;
26:         **if** $upper < ub$ **then**
27:           $ub \leftarrow upper$;
28:           $\bar{\boldsymbol{x}} \leftarrow \boldsymbol{x}^\star$;
29:           remove any node in $\mathcal{N}$ with $lower \geq ub$;
30:         **end if**
31:         **if** $lower < ub$ **then**
32:           add newnode to $\mathcal{N}$;
33:         **end if**
34:     **end for**
35:     remove $(\mathcal{I}_0, \mathcal{I}_1)$ from $\mathcal{N}$;
36:     update lb to be the lowest value of $lower$ over $\mathcal{N}$;
37: **end while**
38: Return $\bar{\boldsymbol{x}}$, lb.

---

validation error has not decreased for a prespecified number of iterations. To illustrate the connection in the case of Algorithm 2, the upper bound loosely plays the role of the validation error while the lower bound loosely plays the role of the training error. Note that the neural network literature suggests an alternate approach to early stopping Algorithm 2 (instead of an explicit time limit) by terminating the algorithm after the upper bound has remained unchanged after visiting some prespecified number of nodes in the enumeration tree.

6 Computational Results

We evaluate the performance of our branch-and-bound algorithm (Algorithm 2, with $\gamma = \sqrt{n}$), our second order cone lower bound (13) (with $\gamma = \sqrt{n}$) and our semidefinite lower bound (21) (with $\gamma = \sqrt{n}$ and $d = 1$) implemented in Julia 1.5.2 using the `JuMP.jl` package version 0.21.7, using `Gurobi` version 9.0.3 to solve all second order cone optimization (sub)problems and using `Mosek` version 9.3 to solve all semidefinite optimization problems. We compare our methods against Basis Pursuit Denoising (BPD) given by (5), Iterative Reweighted ℓ_1 Minimizaton (IRWL1) described in Section 2.2 and Orthogonal Matching Pursuit (OMP) described in Section 2.3. We perform experiments using synthetic data and two real world data sets. We conduct our experiments on MIT's Supercloud Cluster [50], which hosts Intel Xeon Platinum 8260 processors and cores with 4GB RAM. To bridge the gap between theory and practice, we have made our code freely available on GitHub at github.com/NicholasJohnson2020/DiscreteCompressedSensing.jl.

6.1 Synthetic Data Experiments

To evaluate the performance of Algorithm 2, BPD, IRWL1 and OMP on synthetic data, we consider the sparsity of the solution returned by each method, its accuracy (ACC), true positive rate (TPR) and true negative rate (TNR). Let $\boldsymbol{x}^{true} \in \mathbb{R}^n$ denote the ground truth and consider an arbitrary vector $\hat{\boldsymbol{x}} \in \mathbb{R}^n$. Let $\mathcal{I}^{true} = \{i : |x_i^{true}| > 10^{-4}\}$, $\hat{\mathcal{I}} = \{i : |\hat{x}_i| > 10^{-4}\}$. The sparsity of $\hat{\boldsymbol{x}}$ is given by $|\hat{\mathcal{I}}|$. We define the accuracy of $\hat{\boldsymbol{x}}$ as

$$ACC(\hat{\boldsymbol{x}}) = \frac{\sum_{i \in \mathcal{I}^{true}} \mathbb{1}\{|\hat{x}_i| > 10^{-4}\} + \sum_{i \notin \mathcal{I}^{true}} \mathbb{1}\{|\hat{x}_i| \leq 10^{-4}\}}{n}.$$

Similarly, we define the true positive rate of $\hat{\boldsymbol{x}}$ as

$$TPR(\hat{\boldsymbol{x}}) = \frac{\sum_{i \in \mathcal{I}^{true}} \mathbb{1}\{|\hat{x}_i| > 10^{-4}\}}{|\hat{\mathcal{I}}|},$$

and we define the true negative rate of $\hat{\boldsymbol{x}}$ as

$$TNR(\hat{\boldsymbol{x}}) = \frac{\sum_{i \notin \mathcal{I}^{true}} \mathbb{1}\{|\hat{x}_i| \leq 10^{-4}\}}{n - |\hat{\mathcal{I}}|}.$$

To evaluate the performance of (13) and (21), we consider the strength of the lower bound and execution time of each method. We seek to answer the following questions:

1. How does the performance of Algorithm 2 compare to state-of-the-art methods such as BPD, IRWL1 and OMP on synthetic data?
2. How is the performance of Algorithm 2 affected by the number of features n, the underlying sparsity k of the ground truth, and the tolerance parameter ϵ?
3. How does the strength of the lower bound produced by (21) compare to that produced by (13)?

6.1.1 Synthetic Data Generation

To generate synthetic data $\boldsymbol{x} \in \mathbb{R}^n$, $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ and $\boldsymbol{b} \in \mathbb{R}^m$, we first select a random subset of indices $\mathcal{I}^{true} \subset [n]$ that has cardinality k ($|\mathcal{I}^{true}| = k$) and sample $x_i \sim N(0, \frac{\sigma^2}{n})$ for $i \in \mathcal{I}^{true}$ (for $i \notin \mathcal{I}^{true}$, we fix $x_i = 0$). Next, we sample $A_{ij} \sim N(0, \frac{\sigma^2}{n})$ where $\sigma > 0$ is a parameter that controls the signal to noise ratio. We fix $\sigma = 10$ and $m = 100$ throughout all experiments unless stated otherwise. Next, we set $\boldsymbol{b} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{n}$ where $n_j \sim N(0, \sigma^2)$. Finally, we set $\epsilon = \alpha \|\boldsymbol{b}\|_2^2$. $\alpha \in [0, 1]$ is a parameter that can be thought of as controlling the proportion of observations that are allowed to go unexplained by a solution to (3).

6.1.2 Sensitivity to n

We present a comparison of Algorithm 2 with BPD, IRWL1 and OMP as we vary the number of features n. In these experiments, we fixed $k = 10$, and $\alpha = 0.2$ across all trials. We varied $n \in \{100, 200, 300, 400, 500, 600, 700, 800\}$ and we performed 100 trials for each value of n. We give Algorithm 2 a cutoff time of 10 minutes. For IRWL1, we terminate the algorithm after the 50th iteration or after two subsequent iterates are equal up to numerical tolerance. Formally, letting $\bar{\boldsymbol{x}}_t$ denote the iterate after iteration t of IRWL1, we terminate the algorithm if either $t > 50$ or if $\|\bar{\boldsymbol{x}}_t - \bar{\boldsymbol{x}}_{t-1}\|_2 \leq 10^{-6}$. Additionally, we further sparsify the solutions returned by BPD (respectively IRWL1) by performing a greedy rounding following the procedure defined by Algorithm 1 where we pass the solution returned by BPD (respectively IRWL1) as input to the algorithm in place of an optimal solution to (26).

We report the sparsity, accuracy (ACC), true positive rate (TPR) and true negative rate (TNR) for each method in Figure 1. We additionally report the sparsity accuracy and execution time for each method in Tables 1, 2 and 3. The performance metric of greatest interest is the sparsity. Our main findings from this set of experiments are:

1. Algorithm 2 systematically produces sparser solutions than OMP, IRWL1 and BPD. This trend holds in all but one trial (see Table 1). Algorithm 2 on average produces solutions that are 2.71% more sparse than OMP, 16.62% more sparse than BPD and 6.04% more sparse than IRWL1. BPD is the poorest performing method in terms of sparsity of the fitted solutions. We remind the reader that sparsity is computed only after a greedy rounding of the BPD (respectively IRWL1) solution. The sparsity of the BPD (respectively IRWL1) solution prior to rounding is much greater. Indeed, before further sparsifying the BPD (respectively IRWL1) solution,

the solution returned by Algorithm 2 is on average 66.33% (respectively 6.21%) more sparse than the BPD (respectively IRWL1) solution. The sparsity of solutions returned by all methods increases as the number of features n increases.

2. Algorithm 2 marginally outperforms the benchmark methods on accuracy with the exception of the first two parameter configurations ($n = 100$ and $n = 200$, see Table 2). The accuracy of all methods tends to trend upwards with increasing n.

3. The TPR and TNR of all methods are roughly comparable across these experiments. The TPR of all methods decreases while the TNR increases as the number of features n is increased. The sharp drop off in the TPR of all methods as n increases from 100 to 400 is consistent with the number of features selected by each method increasing sharply as n increases from 100 to 400. The latter results in the denominator used in the TPR computation to grow sharply which produces the observed behavior.
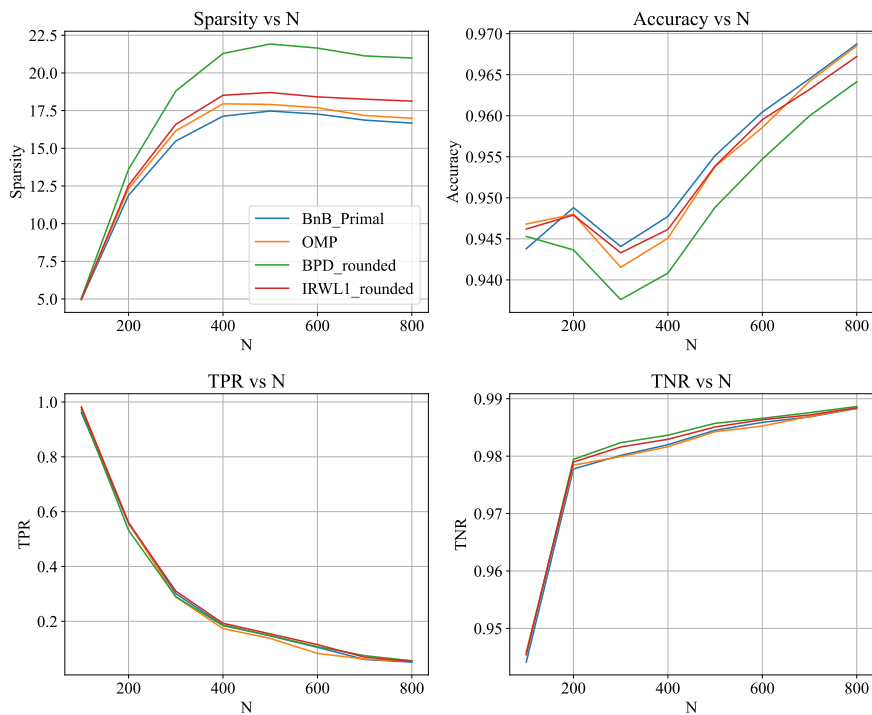

Fig. 1 Sparsity (top left), accuracy (top right), true positive rate (bottom left) and true negative rate (bottom right) versus n with $k = 10$, and $\alpha = 0.2$. Averaged over 100 trials for each parameter configuration.

Table 1 Comparison of the sparsity of solutions returned by (2), OMP, IRWL1 and BPD for different values of n. Averaged over 100 trials for each parameter configuration.

	Sparsity			
N	Algorithm 2	OMP	IRWL1	BPD
100	**5.0**	**5.0**	**5.0**	**5.0**
200	**11.9**	12.3	12.5	13.6
300	**15.5**	16.2	16.6	18.8
400	**17.1**	17.9	18.5	21.3
500	**17.5**	17.9	18.7	21.9
600	**17.3**	17.7	18.4	21.6
700	**16.9**	17.2	18.3	21.1
800	**16.7**	17.0	18.1	21.0

Table 2 Comparison of the accuracy of solutions returned by (2), OMP, IRWL1 and BPD for different values of n. Averaged over 100 trials for each parameter configuration.

	Accuracy			
N	Algorithm 2	OMP	IRWL1	BPD
100	0.944	**0.947**	0.946	0.945
200	**0.949**	0.948	0.948	0.944
300	**0.944**	0.942	0.943	0.938
400	**0.948**	0.945	0.946	0.941
500	**0.955**	0.954	0.954	0.949
600	**0.960**	0.959	**0.960**	0.955
700	**0.965**	0.964	0.963	0.960
800	**0.969**	**0.969**	0.967	0.964

6.1.3 Sensitivity to k

We present a comparison of Algorithm 2 with BPD, IRWL1 and OMP as we vary k the sparsity of the underlying ground truth signal. In these experiments, we fixed $n = 200$ and $\alpha = 0.2$ across all trials. We varied $k \in \{10, 15, 20, 25, 30, 35, 40, 45, 50, 55\}$ and we performed 100 trials for each value of k. We give Algorithm 2 a cutoff time of 10 minutes.

We report the sparsity, accuracy (ACC), true positive rate (TPR) and true negative rate (TNR) for each method in Figure 2. We additionally report the sparsity, accuracy and execution time for each method in Tables 4, 5 and 6. Our main findings from this set of experiments are:

1. Consistent with the results in the previous section, Algorithm 2 systematically produces sparser solutions than OMP, IRWL1 and BPD. This trend holds across trials (see Table 4. Algorithm 2 on average produces solutions that are 4.78% more sparse than OMP, 10.73% more sparse than BPD and 4.20% more sparse than IRWL1. Before further sparsifying the BPD (respectively IRWL1) solution, the

Table 3 Comparison of the execution time of solutions returned by (2), OMP, IRWL1 and BPD for different values of n. Averaged over 100 trials for each parameter configuration.

	Execution Time (milliseconds)			
N	Algorithm 2	OMP	IRWL1	BPD
100	2048.646	**5.717**	463.636	146.111
200	334804.020	**13.212**	1109.263	234.263
300	574501.859	**25.141**	1630.212	297.849
400	601792.939	**42.919**	2181.636	351.717
500	601424.020	**72.535**	2435.141	405.131
600	601451.838	**110.364**	3118.465	433.626
700	601572.848	**166.525**	3674.980	504.626
800	601716.929	**231.980**	3865.788	540.859

solution returned by Algorithm 2 is on average 62.97% (respectively 4.29%) more sparse than the BPD (respectively IRWL1) solution. BPD is again the poorest performing method in terms of sparsity of the fitted solutions. IRWL1 and OMP produce comparably sparse solutions. The sparsity of solutions returned by all methods initially decreases than subsequently increases as the sparsity level k of the ground truth signal increases.

2. Algorithm 2 is competitive with OMP and IRWL1 on accuracy and slightly outperforms BPD on accuracy for larger values of k The accuracy of all methods trends downwards with increasing k, suggesting that the feature identification problem becomes more challenging for larger values of k in this regime.

3. The TPR and TNR of Algorithm 2, OMP, and IRWL1 are comparable across these experiments. The TPR and TNR of BPD is competitive with the other methods for small values of k, but slightly deteriorates for larger values of k.

Table 4 Comparison of the sparsity of solutions returned by (2), OMP, IRWL1 and BPD for different values of k. Averaged over 100 trials for each parameter configuration.

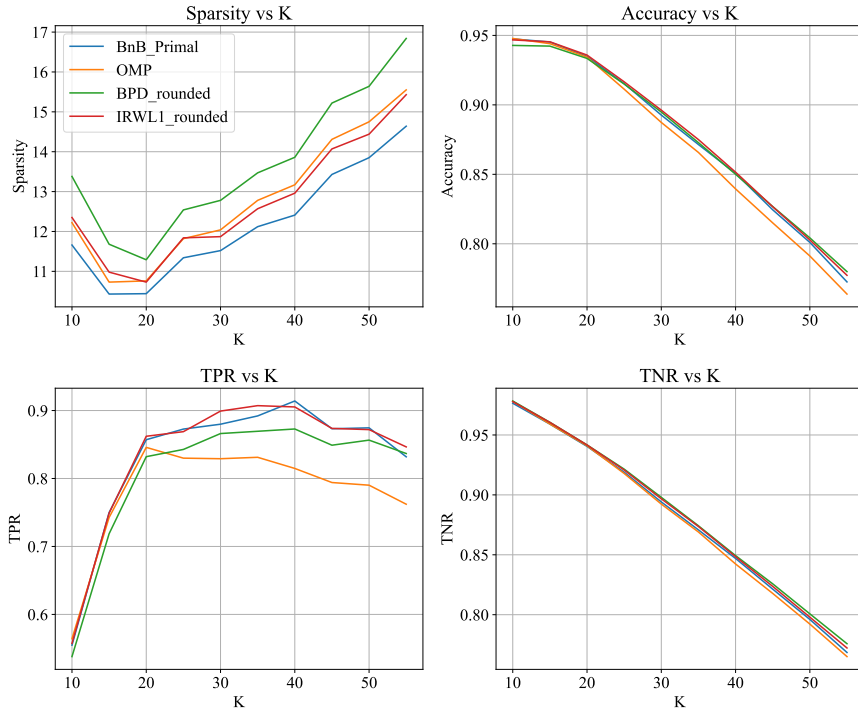|	Sparsity			
K	Algorithm 2	OMP	IRWL1	BPD
10	**11.7**	12.2	12.3	13.4
15	**10.4**	10.7	11.0	11.7
20	**10.4**	10.8	10.7	11.3
25	**11.3**	11.8	11.8	12.5
30	**11.5**	12.0	11.9	12.8
35	**12.1**	12.8	12.6	13.5
40	**12.4**	13.2	13.0	13.9
45	**13.4**	14.3	14.1	15.2
50	**13.8**	14.8	14.4	15.6
55	**14.6**	15.6	15.4	16.8

Fig. 2 Sparsity (top left), accuracy (top right), true positive rate (bottom left) and true negative rate (bottom right) versus k with $N = 200$ and $\alpha = 0.2$. Averaged over 100 trials for each parameter configuration.

6.1.4 Sensitivity to ϵ

We present a comparison of Algorithm 2 with BPD, IRWL1 and OMP as we vary α which controls the value of the parameter ϵ. Recall we have $\epsilon = \alpha\|\boldsymbol{b}\|_2^2$, so α can loosely be interpreted as the fraction of the measurements $\boldsymbol{b}$ that can be unexplained by the returned solution to (3). In these experiments, we fixed $n = 200$ and $k = 10$ across all trials. We varied $\alpha \in \{0.05, 0.1, 0.15, \ldots, 0.9\}$ and we performed 100 trials for each value of α. We give Algorithm 2 a cutoff time of 10 minutes.

We report the sparsity, accuracy (ACC), true positive rate (TPR) and true negative rate (TNR) for each method in Figure 3, and we report the sparsity, accuracy and execution time for each method in Tables 7, 8 and 9. Consistent with previous experiments, Algorithm 2 outperforms the benchmark methods in terms of sparsity of the returned solution while having comparable performance on accuracy, TPR and TNR. Here, Algorithm 2 on average produces solutions that are 2.40% more sparse than OMP, 5.92% more sparse than BPD and 2.54% more sparse than IRWL1. Before further sparsifying the BPD (respectively IRWL1) solution, the solution returned by Algorithm 2 is on average 59.23% (respectively 2.62%) more sparse than the BPD (respectively IRWL1) solution.

Table 5 Comparison of the accuracy of solutions returned by (2), OMP, IRWL1 and BPD for different values of k. Averaged over 100 trials for each parameter configuration.

	Accuracy			
K ‖	Algorithm 2	OMP	IRWL1	BPD
10	**0.948**	**0.948**	0.947	0.943
15	**0.945**	0.944	**0.945**	0.942
20	0.935	0.934	**0.936**	0.933
25	0.915	0.911	**0.917**	0.915
30	0.893	0.887	**0.896**	0.895
35	0.872	0.866	**0.875**	0.873
40	0.851	0.840	**0.852**	0.850
45	0.825	0.815	0.827	**0.827**
50	0.801	0.791	0.803	**0.804**
55	0.772	0.764	0.777	**0.780**

Table 6 Comparison of the execution time of solutions returned by (2), OMP, IRWL1 and BPD for different values of k. Averaged over 100 trials for each parameter configuration.

	Execution Time (milliseconds)			
K ‖	Algorithm 2	OMP	IRWL1	BPD
10	305993.475	**13.182**	1270.000	341.454
15	199128.374	**12.556**	1144.818	284.071
20	119282.667	**12.646**	1080.535	278.596
25	139224.525	**13.263**	1081.202	327.151
30	171844.485	**12.909**	1169.798	314.192
35	193257.535	**12.798**	1163.121	361.485
40	231721.737	**13.404**	1151.455	277.647
45	314269.394	**13.495**	1142.374	308.919
50	351790.071	**13.727**	1219.707	315.081
55	412429.717	**14.010**	1260.899	289.616

6.1.5 Lower Bound Performance

In Section 4, we reformulated (3) exactly as a mixed integer second order cone problem and illustrated multiple approaches to obtain lower bounds on the optimal value of the reformulation. In this Section, we compare the strength of the second order cone relaxation given by (13) and the semidefinite cone relaxation given by (21). We fixed $k = 10$ and we varied $\alpha \in \{0.05, 0.1, 0.15, \ldots, 0.9\}$. We report results for $(n, m) = (25, 100)$ in Figure 4 and $(n, m) = (50, 25)$ in Figure 5. We performed 100 trials for each value of α. Letting lb^{SOC} denote the optimal value of (13) and lb^{SOS} denote the optimal value of (21), we define the SOS lower bound improvement to be $\frac{lb^{SOS} - lb^{SOC}}{lb^{SOC}}$.

Consistent with the Theorem 5, Problem (21) produces a stronger lower bound than Problem (13) at the expense of being more computationally intensive to compute due to the presence of positive semidefinite constraints. On average, the bound produced
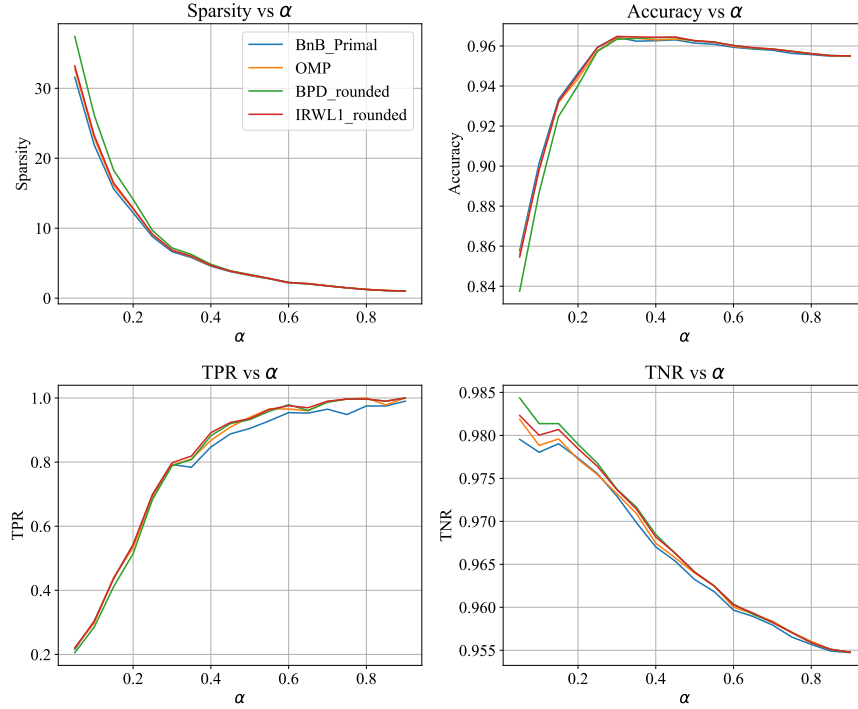
Fig. 3 Sparsity (top left), accuracy (top right), true positive rate (bottom left) and true negative rate (bottom right) versus α with $n = 200$ and $k = 10$. Averaged over 100 trials for each parameter configuration.

by (21) is 8.92% greater than the bound produced by (13). These results suggests that if Problem (21) can be solved to optimality or near optimality efficiently at scale, it could potentially be used to accelerate Algorithm 2 by producing stronger lower bounds than the current approach, thereby allowing for a more aggressive pruning of the feasible space. Off the shelf interior point methods suffer from scalability challenges for semidefinite optimization problems.

6.2 Electrocardiogram Signal Acquisition

We seek to answer the following question: how does the performance of Algorithm 2 compare to state-of-the-art methods such as BPD, IRWL1 and OMP on signal processing using real world data? To evaluate performance, we consider the problem of compressed sensing for electrocardiogram (ECG) acquisition [6]. We obtain real ECG recording samples from the MIT-BIH Arrhythmia Database (https://www.physionet. org/content/mitdb/1.0.0/) and consider the performance of the methods in terms of sparsity of the returned signal and reconstruction error between the returned signal and the true signal.

Table 7 Comparison of the sparsity of solutions returned by (2), OMP, IRWL1 and BPD for different values of α. Averaged over 100 trials for each parameter configuration.

|	Sparsity			
α	Algorithm 2	OMP	IRWL1	BPD
0.05	**31.6**	32.8	33.2	37.4
0.10	**21.9**	22.9	23.3	26.2
0.15	**15.6**	16.1	16.5	18.3
0.20	**12.2**	12.7	12.8	14.1
0.25	**8.8**	9.1	9.1	9.7
0.30	**6.6**	6.9	6.9	7.2
0.35	**5.8**	5.9	6.0	6.2
0.40	**4.6**	4.7	4.7	4.8
0.45	**3.8**	3.9	3.9	3.9
0.50	**3.2**	3.3	3.3	3.4
0.55	**2.8**	2.8	2.8	2.8
0.60	**2.2**	2.2	2.2	2.2
0.65	**2.0**	2.1	2.1	2.1
0.70	**1.7**	1.8	1.8	1.8
0.75	**1.5**	1.5	**1.5**	1.5
0.80	**1.2**	1.3	**1.2**	**1.2**
0.85	**1.1**	1.1	1.1	1.1
0.90	**1.0**	**1.0**	**1.0**	**1.0**
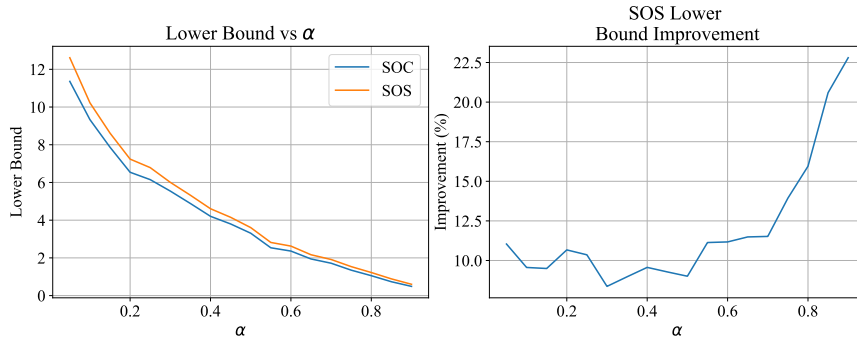
Fig. 4 Problem (3) lower bound (left) produced by Problem (13) (SOC) and Problem (21) (SOS) with $d = 1$. Percent improvement of Problem (3) lower bound of compared to (right). $n = 25, m = 100$ and $k = 10$.

6.2.1 ECG Experiment Setup

We employ the same 100 ECG recordings sampled at 360 Hz from the MIT-BIH Arrhythmia Database that are used in [6]. These recordings collectively originate from 10 distinct patients (each contributing 10 independent recordings) and the recording length of an individual record is 1024. In keeping with [6], we use 30 ECG recordings as a training set to fit an overcomplete dictionary $\boldsymbol{D}$ via the K-SVD method [51]. We fit a dictionary with 2000 atoms, meaning that $\boldsymbol{D} \in \mathbb{R}^{1024 \times 2000}$ and $\boldsymbol{X}^{train} \approx \boldsymbol{D\Theta}$

Table 8 Comparison of the accuracy of solutions returned by (2), OMP, IRWL1 and BPD for different values of α. Averaged over 100 trials for each parameter configuration.

	Accuracy			
α $\parallel$	Algorithm 2	OMP	IRWL1	BPD
0.05	**0.858**	0.856	0.855	0.838
0.10	**0.901**	0.898	0.898	0.887
0.15	**0.933**	0.932	0.932	0.925
0.20	**0.946**	0.944	0.946	0.940
0.25	**0.959**	0.958	0.959	0.957
0.30	0.964	0.964	**0.965**	0.963
0.35	0.962	0.964	**0.965**	0.964
0.40	0.963	0.963	**0.964**	0.964
0.45	0.963	0.963	**0.965**	0.964
0.50	0.962	**0.963**	0.963	0.963
0.55	0.961	**0.962**	0.962	0.962
0.60	0.959	0.960	0.960	**0.960**
0.65	0.959	0.959	**0.959**	0.959
0.70	0.958	**0.959**	0.958	0.958
0.75	0.956	**0.957**	0.957	0.957
0.80	0.956	**0.956**	0.956	0.956
0.85	0.955	0.955	**0.955**	**0.955**
0.90	0.955	**0.955**	**0.955**	**0.955**
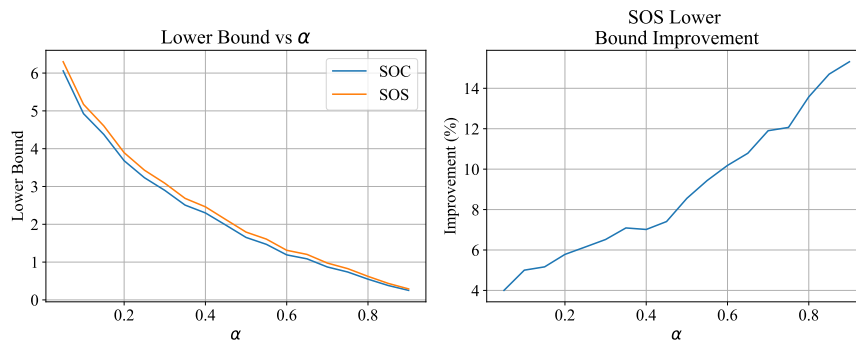
Fig. 5 Problem (3) lower bound (left) produced by Problem (13) (SOC) and Problem (21) (SOS) with $d = 1$. Percent improvement of Problem (3) lower bound of compared to (right). $n = 50, m = 25$ and $k = 10$.

where $\boldsymbol{X}^{train} \in \mathbb{R}^{1024 \times 30}$ is a matrix whose columns are the training ECG signals and $\boldsymbol{\Theta} \in \mathbb{R}^{2000 \times 30}$ is a sparse matrix. Each column of $\boldsymbol{\Theta}$ should be thought of as a (sparse) representation of the corresponding column of $\boldsymbol{X}^{train}$ in the dictionary given by $\boldsymbol{D}$ ($\|\boldsymbol{\Theta}\|_0 \ll \|\boldsymbol{X}^{train}\|_0$). We employ the Bernouilli sensing matrix $\boldsymbol{B} \in \mathbb{R}^{40 \times 1024}$ considered by [6]. Given an ECG signal $\boldsymbol{x}^{test} \in \mathbb{R}^{1024}$, we consider the perturbed observations $\boldsymbol{s} = \boldsymbol{B}(\boldsymbol{x}^{test} + \boldsymbol{\eta})$ where $\boldsymbol{\eta} \in \mathbb{R}^{1024}$ is a vector of mean 0 normal perturbations with variance $\left(\frac{\|\boldsymbol{x}^{test}\|_1}{4 \cdot 1024}\right)^2 \boldsymbol{I}$. Figure 6 illustrates the ECG signal and perturbed ECG signal

Table 9 Comparison of the execution time of solutions returned by (2), OMP, IRWL1 and BPD for different values of α. Averaged over 100 trials for each parameter configuration.

α		Algorithm 2	OMP	IRWL1	BPD
		Execution Time (milliseconds)			
0.05		603205.808	**22.798**	5164.919	934.717
0.10		577124.980	**18.061**	2158.465	522.212
0.15		454366.242	**15.535**	1801.596	636.172
0.20		343487.778	**14.636**	2013.323	967.657
0.25		181672.232	**13.970**	1477.374	654.091
0.30		81074.212	**14.253**	1087.737	510.162
0.35		61929.838	**13.475**	1503.990	904.788
0.40		41775.950	**13.838**	1213.343	684.101
0.45		15927.495	**18.939**	1056.717	531.091
0.50		10387.101	**13.687**	1384.343	872.040
0.55		7032.818	**18.091**	1001.253	530.556
0.60		858.253	**13.212**	1118.929	640.242
0.65		1012.121	**18.404**	1280.869	926.727
0.70		494.808	**18.333**	783.697	514.081
0.75		499.677	**19.899**	696.495	511.717
0.80		749.626	**20.283**	965.010	910.485
0.85		479.091	**20.606**	539.768	537.869
0.90		462.808	**13.394**	485.869	514.566

for record 31 of the dataset. With these preliminaries, we consider the reconstruction problem given by

$$\min_{\boldsymbol{\theta} \in \mathbb{R}^{2000}} \quad \|\boldsymbol{\theta}\|_0 + \frac{1}{\gamma}\|\boldsymbol{\theta}\|_2^2$$
$$\text{s.t.} \qquad \|\boldsymbol{BD\theta} - \boldsymbol{s}\|_2^2 \leq \epsilon. \tag{28}$$

where we set $\epsilon = 1.05 \cdot \|\boldsymbol{s} - \boldsymbol{Bx}^{test}\|_2^2$. Note that (28) is equivalent to (3) where $(\boldsymbol{\theta}, \boldsymbol{BD}, \boldsymbol{s})$ play the role of $(\boldsymbol{x}, \boldsymbol{A}, \boldsymbol{b})$ and we have $(n, m) = (2000, 40)$. Letting $\hat{\boldsymbol{\theta}}$ denote a feasible solution to (28) returned by one of the solution methods, we employ 10^{-4} as the numerical threshold to compute the sparsity $\|\hat{\boldsymbol{\theta}}\|_0$ of $\hat{\boldsymbol{\theta}}$ and we define the ℓ_q reconstruction error of $\hat{\boldsymbol{\theta}}$ as $\frac{\|\boldsymbol{D}\hat{\boldsymbol{\theta}} - \boldsymbol{x}^{test}\|_q^q}{\|\boldsymbol{x}^{test}\|_q^q}$ for $q \in \{1, 2\}$.

6.2.2 ECG Computational Results

We present a comparison of BPD, IRWL1, OMP and Algorithm 2 as we vary the regularization parameter γ in (28). We considered values of γ in the set

$$\Gamma = \{(8a + 0.01)f(n) : a \in [14], f(n) \in \{\sqrt{n}, n, n^2\}\},$$

and we evaluate performance on the 70 ECG recordings that are not part of the training set used to fit the overcomplete dictionary $\boldsymbol{D}$. We give Algorithm 2 a cutoff time of 5 minutes. As in the synthetic experiments, we terminate IRWL1 after the 50th iteration or after two subsequent iterates are equal up to numerical tolerance.
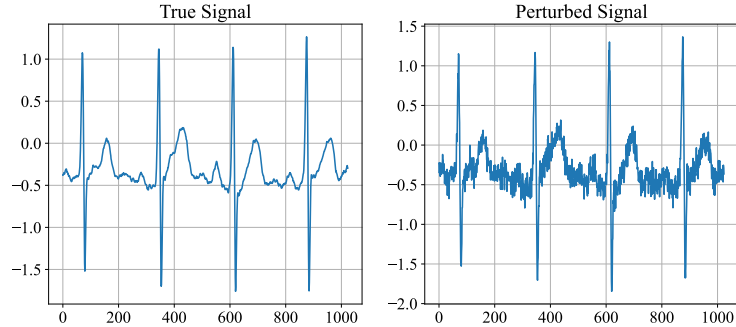
Fig. 6 Ground truth ECG signal (left) and perturbed signal (right) for ECG record 31.

Moreover, we sparsify the solutions returned by BPD and IRWL1 using the procedure given by Algorithm 1 as done in the synthetic experiments.

Figure 7 illustrates the average ℓ_1 error (left) and average ℓ_2 error (right) versus the average sparsity of solutions returned by each method. Each red dot corresponds to the performance of Algorithm 2 for a fixed value of $\gamma \in \Gamma$. Given that more sparse solutions and solutions with lesser ℓ_1 (respectively ℓ_2) error are desirable, Figure 7 demonstrates that as we vary γ, the solutions returned by Algorithm 2 trace out an efficient frontier that dominates the solutions returned by BPD, IRWL1 and OMP. Indeed, for all benchmark methods (BPD, IRWL1 and OMP), there is a value of γ such that Algorithm 2 finds solutions that achieve lower sparsity and lower reconstruction error than the solution returned by the benchmark method. For the same ℓ_2 reconstruction error, Algorithm 2 can produce solutions that are on average 3.88% more sparse than IRWL1, 6.29% more sparse than BPD and 19.70% more sparse than OMP. For the same sparsity level, Algorithm 2 can produce solutions that have on average 1.42% lower ℓ_2 error than IRWL1, 2.66% lower ℓ_2 error than BPD and 28.23% lower ℓ_2 error than OMP. Thus, Algorithm 2 outperforms BPD, IRWL1 and OMP on this real world dataset.

6.3 Multi-Label Classification

We seek to answer the following question: how does the performance of Algorithm 2 compare to state-of-the-art methods such as BPD, IRWL1 and OMP when used as part of a learning task on real world data? To evaluate the performance of the aforementioned algorithms in this setting, we consider the problem of multi-label classification (MLC) [1]. In MLC, given a dataset $\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^n$ where $\boldsymbol{x}_i \in \mathbb{R}^m$ denotes a feature vector and $\boldsymbol{y}_i \in \{0, 1\}^d$ denotes a label vector, we seek to learn a function $f : \mathbb{R}^m \to \{0, 1\}^d$ that can correctly classify unseen examples. Note that this is a more general problem than a typical multi-class classification problem with d classes since in this setting each example can be a member of multiple classes simultaneously. When the number of classes d is very large, training d individual classifiers in a one-against-all approach becomes prohibitively expensive. In this regime, if the label vectors tend to be sparse relative to the size of the latent dimension d, one approach
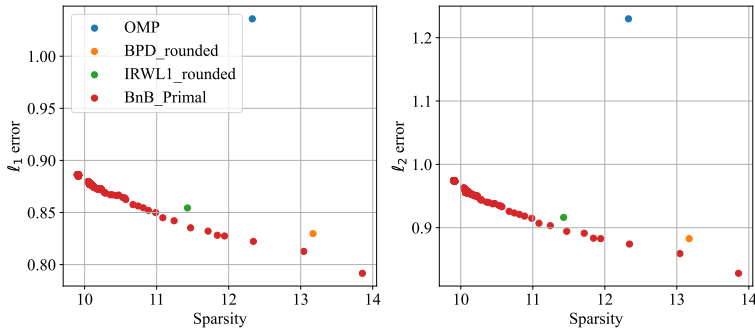
Fig. 7 ℓ_1 reconstruction error (left) and ℓ_2 reconstruction error (right) versus ℓ_0 norm (Sparsity) for ECG reconstructions obtained using OMP, BPD, IRWL1 and Algorithm 2 for varying values of γ. $n = 2000$ and $m = 40$.

is to project the labels into a smaller dimension $k \ll d$ using a linear compression function and then to learn k predictors mapping the feature space to the projected label space. Predictions can then be made by applying the k learned predictors on a given example and subsequently using a compressed sensing based reconstruction algorithm to transform the prediction from projected label space to the latent label space [1, 52–54]. Explicitly, given a linear map $\boldsymbol{P} \in \mathbb{R}^{k \times d}$ where $k \ll d$, we form a transformed dataset $\{(\boldsymbol{x}_i, \boldsymbol{P} \boldsymbol{y}_i)\}_{i=1}^n$ and for each $j \in \{1, \ldots, k\}$ learn a regression function $h_j : \mathbb{R}^m \to \mathbb{R}$ using the data given by $\{(\boldsymbol{x}_i, [\boldsymbol{P} \boldsymbol{y}_i]_j)\}_{i=1}^n$. Given a reconstruction function $g : \mathbb{R}^{k \times d} \times \mathbb{R}^k \to \mathbb{R}^d$ that outputs a sparse vector such that $\boldsymbol{P}[g(\boldsymbol{P}, \boldsymbol{l})] \approx \boldsymbol{l}$, we make predictions on a test point $\boldsymbol{x}_{test}$ by returning $g(\boldsymbol{P}, [h_1(\boldsymbol{x}_{test}), \ldots, h_k(\boldsymbol{x}_{test})])$. In Sections 6.3.1 and 6.3.2, we evaluate the performance of this approach when Algorithm 2, BPD, IRWL1 and OMP are used as the reconstruction function g.

6.3.1 MLC Experiment Setup

We obtain a text data set consisting of web pages and descriptive textual tags from the former social bookmarking service *del.icio.us* that was originally collected by [55]. The dataset consists of roughly 16000 web pages having $d = 983$ unique labels. Each web page on average is associated with 19 labels. The web pages are each represented as a bag-of-words feature vector. Further details can be found in [55]. We use 80% of the data for training and withhold 20% for testing. We fix $k = 100$ during our experiments. We fix the linear map $\boldsymbol{P} = \frac{1}{\sqrt{k}} \boldsymbol{H}$ where $\boldsymbol{H} \in \mathbb{R}^{k \times d}$ is a matrix obtained by choosing k random rows from the $d \times d$ Hadamard matrix in keeping with the approach taken by [1]. We use ridge regression as our base learning algorithm for the regression functions $\{h_j\}_{j=1}^k$ and tune the ridge parameter via leave one out cross validation over the training set. Given a test data point $\boldsymbol{x}_{test} \in \mathbb{R}^m$, we output a predicted label $\hat{\boldsymbol{y}} \in \mathbb{R}^d$ by solving

35

$$\min_{\boldsymbol{y} \in \mathbb{R}^d} \quad \|\boldsymbol{y}\|_0 + \frac{5}{\sqrt{d}}\|\boldsymbol{y}\|_2^2$$
$$\text{s.t.} \quad \|\boldsymbol{P}\boldsymbol{y} - h(\boldsymbol{x}_{test})\|_2^2 \leq \epsilon. \tag{29}$$

where $[h(\boldsymbol{x}_{test})]_j = h_j(\boldsymbol{x}_{test})$ and we set $\epsilon = 0.25 \cdot \|h(\boldsymbol{x}_{test})\|_2^2$. Letting $\hat{\boldsymbol{y}}$ denote a feasible solution to (29) returned by one of the solution methods, we are interested in measuring the accuracy, precision and recall of the support of $\hat{\boldsymbol{y}}$ relative to the true label $\boldsymbol{y}_{test}$. Let $\mathcal{I}^{true} = \{i : [\boldsymbol{y}_{test}]_i = 1\}$ and $\hat{\mathcal{I}} = \{i : |\hat{y}_i| > 10^{-4}\}$. We define the accuracy of $\hat{\boldsymbol{y}}$ as

$$ACC(\hat{\boldsymbol{y}}) = \frac{\sum_{i \in \mathcal{I}^{true}} \mathbb{1}\{|\hat{y}_i| > 10^{-4}\} + \sum_{i \notin \mathcal{I}^{true}} \mathbb{1}\{|\hat{y}_i| \leq 10^{-4}\}}{d}.$$

Similarly, we define the precision of $\hat{\boldsymbol{y}}$ as

$$Precision(\hat{\boldsymbol{y}}) = \frac{\sum_{i \in \mathcal{I}^{true}} \mathbb{1}\{|\hat{y}_i| > 10^{-4}\}}{|\mathcal{I}^{true}|},$$

and we define the recall (true positive rate) of $\hat{\boldsymbol{y}}$ as

$$TPR(\hat{\boldsymbol{y}}) = \frac{\sum_{i \in \mathcal{I}^{true}} \mathbb{1}\{|\hat{y}_i| > 10^{-4}\}}{|\hat{\mathcal{I}}|}.$$

6.3.2 MLC Computational Results

We present a comparison of test set classification performance when each of BPD, IRWL1, OMP and Algorithm 2 are used as the reconstruction algorithm g. We give Algorithm 2 a cutoff time of 10 minutes. As in the synthetic and ECG experiments, we terminate IRWL1 after the 50th iteration or after two subsequent iterates are equal up to numerical tolerance. Moreover, we sparsify the solutions returned by BPD and IRWL1 using the procedure given by Algorithm 1 as done in the synthetic and ECG experiments. Figure 8 illustrates the average accuracy achieved by each method on the test set. We see that that Algorithm 2 achieves slightly greater average accuracy than the benchmark methods. Moreover, Table 10 illustrates the percentage of test examples on which each method has the best performance for accuracy, precision and recall. We see that Algorithm 2 exhibits superior performance than the baseline methods across all metrics of interest.

6.4 Summary of Findings

We now summarize our findings from our numerical experiments. In Sections 6.1.2-6.1.4, we see that across all experiments using synthetic data, Algorithm 2 produces solutions that are on average 6.22% more sparse than the solutions returned by state of the art benchmark methods after they are further sparsified by greedy rounding. If we omit greedy rounding, Algorithm 2 produces solutions that are on average 17.17% more sparse in our synthetic experiments. In Section 6.1.5, we find that the bound
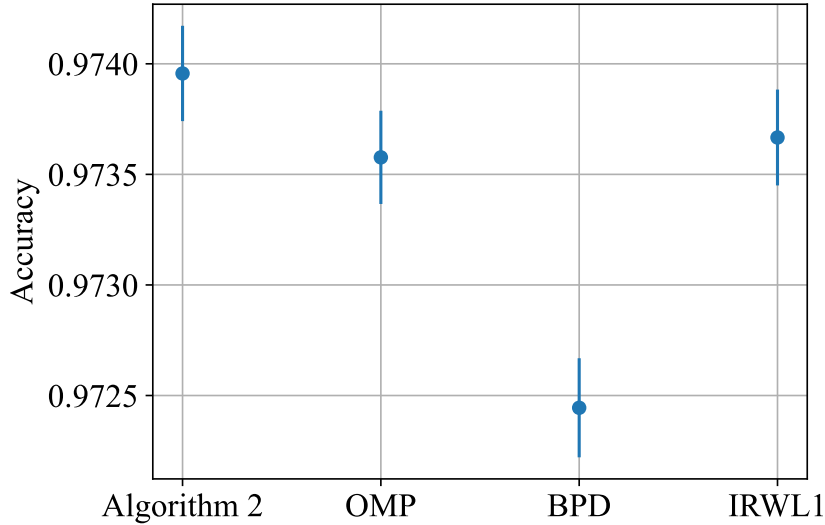
Fig. 8 Test set prediction accuracy obtained using OMP, BPD, IRWL1 and Algorithm 2 as the label reconstruction algorithm.

Table 10 Comparison of the accuracy of solutions returned by (2), OMP, IRWL1 and BPD for different values of α. Averaged over 100 trials for each parameter configuration.

		Top Performing Algorithm Frequency (%)			
Metric	‖	Algorithm 2	OMP	IRWL1	BPD
Accuracy	‖	**53.47**	28.78	11.66	6.09
Precision	‖	**49.39**	21.42	15.26	13.93
Recall	‖	**54.90**	14.14	3.20	27.76

produced by (21) is on average 8.92% greater than the bound produced by (13). In Section 6.2, we see that for a given level of ℓ_2 reconstruction error, Algorithm 2 produces solutions that are on average 9.95% more sparse than the solutions returned by state of the art benchmark methods after they are further sparsified by greedy rounding on the real world ECG dataset we experiment with. Furthermore, for a given sparsity level, Algorithm 2 produces solutions that have on average 10.77% lower ℓ_2 reconstruction error than benchmark methods. Finally, in Section 6.3, we see that Algorithm 2 outperforms benchmark methods when used as the reconstruction algorithm for compressed sensing multi-label classification in terms of accuracy, precision and recall.

7 Conclusion

In this paper, we introduced an ℓ_2 regularized formulation (3) for CS which emits a natural reformulation as a mixed integer second order cone program (12). We presented a second order cone relaxation (13) and a stronger but more expensive semidefinite cone relaxation (21) to (12). We presented Algorithm 2, a custom branch-and-bound algorithm that can compute globally optimal solution for (3). We find that our approach produces solutions that are on average 6.22% more sparse on synthetic data and 9.95% more sparse on real world ECG data when compared to three state of the art benchmark approaches. This improvement in accuracy comes at the cost of an increase in computation time by several orders of magnitude. When comparing only against the experiment-wise best performing benchmark method, our approach produces solutions that are on average 3.10% more sparse on synthetic data and 3.88% more sparse on real world ECG data. Moreover, our approach outperforms benchmark methods when used as part of a multi-label learning algorithm. Further work might focus on strengthening our convex relaxations by deriving additional valid inequalities for (12) or increasing the scalability of our branch-and-bound method. Algorithm 2 currently uses our second order cone relaxation to compute lower bounds. If fast problem specific solution methods could be derived for our positive semidefinite cone relaxation, employing the latter for lower bounds in Algorithm 2 could potentially lead to important scalability gains.

Declarations

Code availability:

To bridge the gap between theory and practice, we have made our code freely available on GitHub at github.com/NicholasJohnson2020/DiscreteCompressedSensing.jl

Authors' contributions:

Both authors contributed to the algorithmic ideation and design. Algorithm implementation, data collection, simulation and data analysis was performed by Nicholas Johnson. The first draft of the manuscript was written by Nicholas Johnson and both authors commented and edited subsequent versions of the manuscript. Both authors read and approved the final manuscript.

References

[1] Hsu, D.J., Kakade, S.M., Langford, J., Zhang, T.: Multi-label prediction via compressed sensing. Advances in neural information processing systems **22** (2009)

[2] Lustig, M., Donoho, D., Pauly, J.M.: Sparse mri: The application of compressed sensing for rapid mr imaging. Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine **58**(6), 1182–1195 (2007)

[3] Brady, D.J., Choi, K., Marks, D.L., Horisaki, R., Lim, S.: Compressive holography. Optics express **17**(15), 13040–13049 (2009)

[4] Hashemi, A., Rostami, M., Cheung, N.-M.: Efficient environmental temperature monitoring using compressed sensing. In: 2016 Data Compression Conference (DCC), pp. 602–602 (2016). IEEE Computer Society

[5] Wang, G., Zhao, Z., Ning, Y.: Design of compressed sensing algorithm for coal mine iot moving measurement data based on a multi-hop network and total variation. Sensors **18**(6), 1732 (2018)

[6] Chen, J., Xing, J., Zhang, L.Y., Qi, L.: Compressed sensing for electrocardiogram acquisition in wireless body sensor network: A comparative analysis. International Journal of Distributed Sensor Networks **15**(7), 1550147719864884 (2019)

[7] Donoho, D.L.: Compressed sensing. IEEE Transactions on information theory **52**(4), 1289–1306 (2006)

[8] Tropp, J.A., Wright, S.J.: Computational methods for sparse solution of linear inverse problems. Proceedings of the IEEE **98**(6), 948–958 (2010)

[9] Rani, M., Dhok, S.B., Deshmukh, R.B.: A systematic review of compressive sensing: Concepts, implementations and applications. IEEE access **6**, 4875–4894 (2018)

[10] Owen, A.B., Perry, P.O.: Bi-cross-validation of the SVD and the nonnegative matrix factorization. The Annals of Applied Statistics **3**(2), 564–594 (2009)

[11] Bousquet, O., Elisseeff, A.: Stability and generalization. Journal of Machine Learning Research **2**, 499–526 (2002)

[12] Karahanoğlu, N.B., Erdoğan, H., Birbil, Ş.İ.: A mixed integer linear programming formulation for the sparse recovery problem in compressed sensing. In: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 5870–5874 (2013). IEEE

[13] Bourguignon, S., Ninin, J., Carfantan, H., Mongeau, M.: Exact sparse approximation problems via mixed-integer programming: Formulations and computational

performance. IEEE Transactions on Signal Processing **64**(6), 1405–1419 (2015)

[14] Chen, S., Donoho, D.: Basis pursuit. In: Proceedings of 1994 28th Asilomar Conference on Signals, Systems and Computers, vol. 1, pp. 41–44 (1994). IEEE

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

[16] Candes, E.J., Tao, T.: Near-optimal signal recovery from random projections: Universal encoding strategies? IEEE transactions on information theory **52**(12), 5406–5425 (2006)

[17] Gill, P.R., Wang, A., Molnar, A.: The in-crowd algorithm for fast basis pursuit denoising. IEEE Transactions on Signal Processing **59**(10), 4595–4605 (2011)

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

[19] Bertsimas, D., Copenhaver, M.S.: Characterization of the equivalence of robustification and regularization in linear and matrix regression. European Journal of Operational Research **270**(3), 931–942 (2018)

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

[21] Donoho, D.L., Elad, M.: Optimally sparse representation in general (nonorthogonal) dictionaries via l1 minimization. Proceedings of the National Academy of Sciences **100**(5), 2197–2202 (2003)

[22] Gribonval, R., Nielsen, M.: Sparse representations in unions of bases. IEEE transactions on Information theory **49**(12), 3320–3325 (2003)

[23] Tropp, J.A.: Greed is good: algorithmic results for sparse approximation. IEEE Transactions on Information Theory **50**(10), 2231–2242 (2004)

[24] Candes, E.J., Tao, T.: Decoding by linear programming. IEEE transactions on information theory **51**(12), 4203–4215 (2005)

[25] Baraniuk, R., Davenport, M., DeVore, R., Wakin, M.: A simple proof of the restricted isometry property for random matrices. Constructive Approximation **28**(3), 253–263 (2008)

[26] Guédon, O., Litvak, A.E., Pajor, A., Tomczak-Jaegermann, N.: Restricted isometry property for random matrices with heavy-tailed columns. Comptes Rendus Mathematique **352**(5), 431–434 (2014)

[27] Candes, E.J., Wakin, M.B., Boyd, S.P.: Enhancing sparsity by reweighted l1

minimization. Journal of Fourier analysis and applications **14**(5), 877–905 (2008)

[28] Needell, D.: Noisy signal recovery via iterative reweighted l1-minimization. In: 2009 Conference Record of the Forty-Third Asilomar Conference on Signals, Systems and Computers, pp. 113–117 (2009). IEEE

[29] Asif, M.S., Romberg, J.: Fast and accurate algorithms for re-weighted l1-norm minimization. IEEE Transactions on Signal Processing **61**(23), 5905–5916 (2013)

[30] Chen, X., Zhou, W.: Convergence of reweighted l1 minimization algorithms and unique solution of truncated lp minimization. Department of Applied Mathematics, The Hong Kong Polytechnic University (2010)

[31] Wang, H., Zhang, F., Shi, Y., Hu, Y.: Nonconvex and nonsmooth sparse optimization via adaptively iterative reweighted methods. Journal of Global Optimization **81**(3), 717–748 (2021)

[32] Wang, H., Zeng, H., Wang, J.: An extrapolated iteratively reweighted l1 method with complexity analysis. Computational Optimization and Applications, 1–31 (2022)

[33] Pati, Y.C., Rezaiifar, R., Krishnaprasad, P.S.: Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. In: Proceedings of 27th Asilomar Conference on Signals, Systems and Computers, pp. 40–44 (1993). IEEE

[34] Mallat, S.G., Zhang, Z.: Matching pursuits with time-frequency dictionaries. IEEE Transactions on signal processing **41**(12), 3397–3415 (1993)

[35] Cai, T.T., Wang, L.: Orthogonal matching pursuit for sparse signal recovery with noise. IEEE Transactions on Information theory **57**(7), 4680–4688 (2011)

[36] Wang, J.: Support recovery with orthogonal matching pursuit in the presence of noise. IEEE Transactions on Signal processing **63**(21), 5868–5877 (2015)

[37] Dai, W., Milenkovic, O.: Subspace pursuit for compressive sensing signal reconstruction. IEEE transactions on Information Theory **55**(5), 2230–2249 (2009)

[38] Günlük, O., Linderoth, J.: Perspective reformulation and applications. In: Mixed Integer Nonlinear Programming, pp. 61–89. Springer, USA (2012)

[39] Lasserre, J.B.: An explicit exact sdp relaxation for nonlinear 0-1 programs. In: Integer Programming and Combinatorial Optimization: 8th International IPCO Conference Utrecht, The Netherlands, June 13–15, 2001 Proceedings 8, pp. 293–303 (2001). Springer

[40] Lasserre, J.B.: Global optimization with polynomials and the problem of moments. SIAM Journal on optimization **11**(3), 796–817 (2001)

[41] Lasserre, J.B.: Moments, Positive Polynomials and Their Applications vol. 1. World Scientific, France (2009)

[42] Boyd, S., Boyd, S.P., Vandenberghe, L.: Convex Optimization. Cambridge university press, USA (2004)

[43] Land, A.H., Doig, A.G.: An Automatic Method for Solving Discrete Programming Problems, pp. 105–132. Springer, Berlin, Heidelberg (2010)

[44] Little, J.D.: Branch and bound methods for combinatorial problems. (1966)

[45] Petersen, K.B., Pedersen, M.S., *et al.*: The matrix cookbook. Technical University of Denmark **7**(15), 510 (2008)

[46] Bertsimas, D., Cory-Wright, R., Johnson, N.A.: Sparse plus low rank matrix decomposition: A discrete optimization approach. The Journal of Machine Learning Research **24**(1), 12478–12528 (2023)

[47] Morrison, D.R., Jacobson, S.H., Sauppe, J.J., Sewell, E.C.: Branch-and-bound algorithms: A survey of recent advances in searching, branching, and pruning. Discrete Optimization **19**, 79–102 (2016)

[48] Bertsimas, D., Digalakis Jr, V.: The backbone method for ultra-high dimensional sparse machine learning. Machine Learning **111**(6), 2161–2212 (2022)

[49] Goodfellow, I.J., Bengio, Y., Courville, A.: Deep Learning. MIT Press, Cambridge, MA, USA (2016)

[50] Reuther, A., Kepner, J., Byun, C., Samsi, S., Arcand, W., Bestor, D., Bergeron, B., Gadepally, V., Houle, M., Hubbell, M., Jones, M., Klein, A., Milechin, L., Mullen, J., Prout, A., Rosa, A., Yee, C., Michaleas, P.: Interactive supercomputing on 40,000 cores for machine learning and data analysis. In: 2018 IEEE High Performance Extreme Computing Conference (HPEC), pp. 1–6 (2018). IEEE

[51] Aharon, M., Elad, M., Bruckstein, A.: K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. IEEE Transactions on signal processing **54**(11), 4311–4322 (2006)

[52] Kapoor, A., Viswanathan, R., Jain, P.: Multilabel classification using bayesian compressed sensing. Advances in neural information processing systems **25** (2012)

[53] Som, S.: Learning label structure for compressed sensing based multilabel classification. In: 2016 SAI Computing Conference (SAI), pp. 54–60 (2016). IEEE

[54] Kai, W., Jingzhi, L., Yanjun, C.: Compressed sensing based multi-label classification without label sparsity level prior. In: Proceedings of the 2017 International Conference on Deep Learning Technologies, pp. 66–69 (2017)

[55] Tsoumakas, G., Katakis, I., Vlahavas, I.: Effective and efficient multilabel classification in domains with large number of labels. In: Proc. ECML/PKDD 2008 Workshop on Mining Multidimensional Data (MMD'08), vol. 21, pp. 53–59 (2008)