Noname manuscript No.
(will be inserted by the editor)

A Primal-Dual Homotopy Algorithm for ℓ_1-Minimization with ℓ_∞-Constraints

Christoph Brauer · Dirk A. Lorenz · Andreas M. Tillmann

Received: date / Accepted: date

Abstract In this paper we propose a primal-dual homotopy method for ℓ_1-minimization problems with infinity norm constraints in the context of sparse reconstruction. The natural homotopy parameter is the value of the bound for the constraints and we show that there exists a piecewise linear solution path with finitely many break points for the primal problem and a respective piecewise constant path for the dual problem. We show that by solving a small linear program, one can jump to the next primal break point and then, solving another small linear program, a new optimal dual solution is calculated which enables the next such jump in the subsequent iteration. Using a theorem of the

This material was based upon work partially supported by the National Science Foundation under Grant DMS-1127914 to the Statistical and Applied Mathematical Sciences Institute. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

Parts of this work were carried out while the third author was with the AG Optimierung at TU Darmstadt.

Christoph Brauer
Technische Universität Braunschweig, Institut für Analysis und Algebra
Universitätsplatz 2, 38106 Braunschweig, Germany
Tel.: +49-531-3917421
Fax: +49-531-3917414
E-mail: ch.brauer@tu-braunschweig.de

Dirk A. Lorenz
Technische Universität Braunschweig, Institut für Analysis und Algebra
Universitätsplatz 2, 38106 Braunschweig, Germany
E-mail: d.lorenz@tu-braunschweig.de

Andreas M. Tillmann
RWTH Aachen University, Visual Computing Institute & Chair of Operations Research
Lehrstuhl für Informatik 8, 52056 Aachen, Germany
E-mail: andreas.tillmann@cs.rwth-aachen.de

alternative, we show that the method never gets stuck and indeed calculates the whole path in a finite number of steps.

Numerical experiments demonstrate the effectiveness of our algorithm. In many cases, our method significantly outperforms commercial LP solvers; this is possible since our approach employs a sequence of considerably simpler auxiliary linear programs that can be solved efficiently with specialized active-set strategies.

Keywords Convex Optimization · Dantzig Selector · Homotopy Methods · Nonsmooth Optimization · Primal-Dual Methods

Mathematics Subject Classification (2000) 90C05 · 90C25 · 65K05

1 Introduction

With the advent of Compressed Sensing [13,12,14,25], recovery of sparse vectors by means of the popular Basis Pursuit approach [10],

$$\min_{x\in\mathbb{R}^n} \|x\|_1 \quad \text{s.t.} \quad Ax = b, \tag{BP}$$

and the so-called Basis Pursuit Denoising (or ℓ_1-regularized Least-Squares) problem

$$\min_{x\in\mathbb{R}^n} \lambda\|x\|_1 + \tfrac{1}{2}\|Ax-b\|_2^2, \tag{ℓ_1-LS}$$

with $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$ and $\lambda > 0$, received a lot of attention both theoretically and algorithmically over the past decade (see, e.g., [18,25] and many references therein). However, the related problem

$$\min_{x\in\mathbb{R}^n} \|x\|_1 \quad \text{s.t.} \quad \|Ax-b\|_\infty \le \delta \tag{P_δ}$$

appears to be much less investigated. This problem can be rewritten as a linear program (LP) by formulating the ℓ_∞-norm constraint as linear inequalities and performing the usual variable split of x into its positive and negative parts (see (1) below). Thus, in principle, every LP solver can be applied to solve the problem. However, in practice it may happen that the problem instances are very large (and with A dense or perhaps only available implicitly) so that current LP solvers may not be able to handle the problem well. Moreover, there are cases in which one does not only want to solve the problem for a given instance of (A, b, δ) but for a whole range of parameters δ.

Our interest in sparse approximation under ℓ_∞-constraints via the problem (P_δ) is motivated by several practical applications:

– The *Dantzig selector* problem [9]

$$\min_{x\in\mathbb{R}^n} \|x\|_1 \quad \text{s.t.} \quad \left\|A^\top(Ax-b)\right\|_\infty \le \delta \tag{DS_δ}$$

is a special case of (P_δ) and has numerous applications in statistical estimation, see, e.g., [30], where the whole solution path for $\delta > 0$ is computed as a selection step prior to a classification step.

- In *sparse dequantization*, one has quantized measurements $b = Q(A\bar{x})$ of some signal vector $\bar{x}$ which is assumed to be sparse. If the quantization level is known, one can interpret (P_δ) as the problem of finding a reconstruction x^* with minimal ℓ_1-norm for which the measurements Ax^* produce the same quantized measurements b. We refer to [15] for the general idea and to [6] for a recent application to speech processing.
- In *sparse linear discriminant analysis* as proposed in [7], one obtains a problem of the form (P_δ) in which A is a sample covariance matrix and b is a difference of sampled means. Similarly, the so-called CLIME estimator [8] solves *sparse precision matrix estimation* problems via a sequence of (P_δ) problems in each of which A is again a covariance matrix and b is equal to a unit vector.

In this paper, we develop a homotopy algorithm for the problem (P_δ). The starting point is that for $\delta \geq \|b\|_\infty$, the vector $x = 0$ is obviously the optimal solution. Moreover, we will show that for a solution x of (P_δ) for a given $\delta > 0$, there exist a direction d and a scalar $t_0 > 0$ such that $x + td$ is a solution of $(\mathrm{P}_{\delta-t})$ for $0 \leq t \leq t_0$. Our algorithm builds on these observations and calculates a path of solutions for decreasing values of δ until a target δ-value is reached; we shall prove that the algorithm is able to compute such a path in finitely many steps (even if the final value is $\delta = 0$). Our approach resembles the popular homotopy method for (ℓ_1-LS), cf. [22], but, as detailed later, our method has to work on both the primal and dual problem simultaneously, so that the algorithms differ considerably.

The remainder of this paper is structured as follows: We further touch upon related methods in Subsection 1.1 below, and fix some notation in Subsection 1.2. The main part of the paper, Section 2, constitutes a detailed derivation of our homotopy approach to (P_δ), including theoretical results on iterative improvement and finite termination. An efficient solution approach for subproblems encountered in our scheme is put forth in Section 3. We consider some practical applications and present computational results in Section 4, discuss possible extensions and conclude the paper in Section 5.

1.1 Related Work

Homotopy concepts have been around for decades, so it should come as no surprise that our approach bears some resemblance to several earlier algorithms. In the following, we briefly comment on similarities and differences with respect to the arguably most naturally related methods.

1.1.1 Parametric Simplex Method

It is well-known that problem (P_δ) can be recast as an LP, e.g.,

$$\begin{aligned} \min_{x^\pm \in \mathbb{R}^n, s^\pm \in \mathbb{R}^m} \quad & \mathbb{1}^\top x^+ + \mathbb{1}^\top x^- \\ \text{s.t.} \quad & \begin{pmatrix} A & -A & I & 0 \\ -A & A & 0 & I \end{pmatrix} \begin{pmatrix} x^+ \\ x^- \\ s^+ \\ s^- \end{pmatrix} = \begin{pmatrix} b + \delta\mathbb{1} \\ -b + \delta\mathbb{1} \end{pmatrix} \\ & x^+,\ x^-,\ s^+,\ s^- \geq 0. \end{aligned} \tag{1}$$

There exists a variety of homotopy schemes for LPs, see, for instance, [5, 20] and references therein. In fact, the latter work shows how many standard LP algorithms (simplex, affine-scaling and interior-point methods) can be subsumed under a unifying homotopy framework, exhibiting nice connections between intuitively very different approaches. The LP homotopy method most naturally related to our approach results from treating the parameter δ itself as the homotopy parameter (as we shall also do in our method) in the above LP—the so-called (self-dual) *parametric simplex method* (PSM) [11, 29]. Very briefly, PSM perturbs both the LP right-hand side and objective coefficient vectors using the same parameter and then drives this parameter down to zero, performing primal or dual simplex pivot steps at each breakpoint in the (piecewise linear) parameter homotopy path. For a sufficiently large initial parameter, a primal-dual feasible (hence, optimal) basis is easily found and used to start the algorithm; reducing the parameter, basis optimality is maintained until either a basic variable or nonbasic reduced cost coefficient changes sign, which identifies the breakpoints and induces an appropriate simplex step to exchange some basis element for a nonbasic one. (For a detailed formal description, see, e.g., [29, pp. 115–121].)

In fact, PSM was very recently proposed for sparse linear discriminant analysis problems by means of reformulating the associated problem (P_δ) as precisely the LP stated above, see [23], in which PSM is applied to several other problems as well. For the above special parameterized LP, one needs to stop PSM as soon as the parameter drops below the target original δ (*not* zero) and since the objective is unperturbed, only primal simplex pivot steps are performed throughout the entire algorithmic process (i.e., each breakpoint identifies some variable that is to leave the basis in exchange for a nonbasic one; neither of these facts is mentioned in [23]).

If the optimal solutions for each respective parameter interval are unique, then PSM and our approach necessarily produce the same solution path. However, the paths may differ if multiple optimal solutions occur, as the underlying algorithmic concepts are different: For one thing, we operate in the original variable space (n primal and m dual variables versus $2n + 2m$ variables in the above parameterized LP), and thus avoid doubling the dimensions. Moreover, in each iteration, PSM is restricted to moving to an adjacent basis and, in particular, can get "stuck" at a certain parameter value for several iterations

(namely when several pivot steps are needed to eventually arrive at a new basis that allows to further reduce the parameter). Such a situation can never occur in our algorithm (cf. Theorem 2.2 in Section 2.4 below); indeed, our scheme guarantees the largest reduction of δ in every iteration and moves directly to associated optimal points.

Regarding implementation, PSM is subject to all advantages and drawbacks that come with any simplex method, e.g., its basic version (as described in [29]) may cycle and hence not even terminate, special care needs to be taken to compute and maintain numerically stable basis matrix factorizations, etc. Our approach is straightforward to implement, but requires access to an LP solver for subproblem optimization—given the large selection of sophisticated LP solvers (both proprietary and freely available) to choose from, we actually consider this a feature, not a disadvantage. In particular, this allows us to use a certain active-set LP strategy that turns out to be particularly well-suited to the subproblems occurring during our method, see Section 3. At least in case of multiple optimal solutions, both PSM and our homotopy method are naturally influenced by choices made for crucial steps (i.e., pivoting rules for PSM and LP subproblem solver choice in our implementation), which makes a direct numerical comparison somewhat meaningless; hence, we do not delve into this subject further. (It should however be noted that the homotopy approach not only provides the whole solution path, but for sparse solutions is also significantly faster than applying a standard LP solver to the LP reformulation of (P_δ) directly.)

Finally, let us remark that the relationship between (P_δ) and linear programming extends, in a sense, both ways: Obviously, a general LP method can be used to solve (P_δ), rewritten as the above LP, but a relevant and relatively large subclass of LPs can also be recast into a form resembling (P_δ) for which our algorithm can be adapted straightforwardly, cf. Section 5.

1.1.2 Dantzig Selector and ℓ_1-Regularized Least-Squares Homotopy

A homotopy scheme for the Dantzig selector problem (DS_δ) was proposed in [3]. There, the general idea is also to perform primal and dual update steps in each iteration, starting from a large value for the parameter δ (for which the optimal solution is trivially known) and driving it down toward the desired level. The update steps consist of finding directions along which optimality conditions are maintained and by choosing suitable step sizes, breakpoints in the homotopy path are identified; the supports of the current primal and dual variables are updated one element at a time[1].

[1] The description in [3] is a bit unclear in this regard; it seems the authors implicitly use a kind of subproblem uniqueness assumption under which this works out well, although the choice of indices entering or leaving a support apparently needs not be uniquely determined in general. Also, they claim the optimality conditions they work with imply uniqueness, but they are equivalent to the standard LP optimality conditions with strict complementary slackness (see, e.g., [26, Section 7.9]) applied to the LP obtainable from (DS_δ), which do not import a statement about uniqueness.

Clearly, (DS_δ) is a special case of the more general problem (P_δ) we consider. Moreover, we allow primal and dual supports to change by more than one component per iteration (and we do not make any uniqueness assumptions), so our approach also generalizes that of [3] conceptually. Another difference is that we do not explicitly compute directions first but directly obtain the respective next points. Nevertheless, the method from [3] remains of interest in its own right, since the special (Gramian) structure of the constraint matrix allows for a more direct subproblem treatment than the LPs we will solve.

As discussed in [16,4], for certain sparsity levels of the optimal solution to (DS_δ) and/or conditions on the matrix A, the whole respective solution paths of the Dantzig selector homotopy from [3], the related but different DASSO algorithm from [16], and the homotopy scheme for (ℓ_1-LS) (see [22]) coincide. (Also, the Dantzig selector homotopy algorithm can be modified quite simply to reduce to the ℓ_1-LS homotopy scheme, cf. [3]).

Thus, our algorithm is naturally related to those methods as well: Though (P_δ) generalizes (DS_δ), which in turn is sometimes equivalent to (ℓ_1-LS), neither problems are equivalent, whence the various algorithms are necessarily different, though certainly very similar in spirit. It is also worth noting that while the homotopy for (ℓ_1-LS) is a primal method[2], the approaches for (DS_δ) and also our proposed algorithm work in a primal-dual fashion.

1.2 Notation

For $A \in \mathbb{R}^{m\times n}$, $a_i^\top$ denotes the i-th row and A_j denotes the j-th column of A. Moreover, for $I \subseteq \{1,\dots,m\}$ and $J \subseteq \{1,\dots,n\}$, A_J^I denotes the sub-matrix of A with rows indicated by I and columns indicated by J. Sometimes, we write $A_J^\top = (A_J)^\top$.

By $\odot$, we denote the component-wise product of two vectors, i.e., for $x, z \in \mathbb{R}^n$, we have $(x \odot z)_j = x_j z_j$.

Furthermore, we define $\mathrm{Diag}(x)$ to be the $n \times n$ diagonal matrix having the entries of the vector x as its diagonal elements.

As usual, $\|\cdot\|_1$ and $\|\cdot\|_\infty$ denote the respective norms, i.e.,

$$\|x\|_1 = \sum_{j=1}^{n} |x_j| \quad \text{and} \quad \|x\|_\infty = \max_{j=1,\dots,n} |x_j|.$$

The subdifferential of $\|\cdot\|_1$ at x is denoted by

$$\mathrm{Sign}(x) \coloneqq \partial\|\cdot\|_1(x) = \big\{\xi \in [-1,1]^n : x_j \neq 0 \Rightarrow \xi_j = \mathrm{sign}(x_j)\big\}.$$

[2] More precisely, due to the smooth ℓ_2-part in (ℓ_1-LS), for every primal optimal solution w.r.t. some parameter δ, the associated dual optimal solution is known in closed-form, which can be substituted into the algorithmic formluae directly, eliminating the need for keeping a dual variable explicitly.

Finally, for given primal variable x, dual variable y and bound δ, we introduce the index sets

$$\begin{aligned} J_P &\coloneqq \{j : x_j \neq 0\} && \text{(primal support)},\\ I_P &\coloneqq \left\{i : \left|a_i^\top x - b_i\right| = \delta\right\} && \text{(primal active set of constraints)},\\ J_D &\coloneqq \left\{j : \left|A_j^\top y\right| = 1\right\} && \text{(dual active set)}\\ \text{and}\quad I_D &\coloneqq \{i : y_i \neq 0\} && \text{(dual support)}, \end{aligned}$$

cf. (P_δ) and its dual problem (D_δ) (defined below). Generally, for notational simplicity, we do not make the sets' dependency on x, y and δ explicit as it will be clear from the context. Nevertheless, if we consider these index sets for specific algorithmic iterates x^k, y^k and δ^k, we write J_P^k, I_P^k, J_D^k, and I_D^k, respectively.

Set complements are denoted by a superscript c and always pertain to the respective natural superset; e.g., $J_P^c = \{1,\dots,n\}\backslash J_P$ and $I_P^c = \{1,\dots,m\}\backslash I_P$.

2 Homotopy Algorithm

In the following, we describe our algorithmic approach in detail and prove its correctness and finite convergence. A pseudocode of the method is given in Algorithm 1 below. With a wink and a nod to a certain well-known basis pursuit solver, we call our algorithm ℓ_1-Houdini (ℓ_1-norm HOmotopy UnDer Infinity-Norm constraInts). Throughout, we assume w.l.o.g. that $\delta < \|b\|_\infty$ (otherwise, $x^* = 0$ trivially solves (P_δ)).

2.1 Optimality Conditions and Algorithmic Idea

By Fenchel-Rockafellar Duality (see, e.g., [24, Theorem 31.2]), it holds that x^* is an optimal solution of (P_δ) if and only if there exists a y^* such that

$$-A^\top y^* \in \operatorname{Sign}(x^*) \tag{2}$$

$$\text{and}\quad Ax^* - b \in \delta \operatorname{Sign}(y^*). \tag{3}$$

In particular, such a y^* is an optimal solution to the dual problem of (P_δ), i.e.,

$$\max_{y\in\mathbb{R}^m} \; -b^\top y - \delta\|y\|_1 \quad \text{s.t.} \;\; \left\|A^\top y\right\|_\infty \leq 1. \tag{D_δ}$$

Thus, we call y^* a *dual certificate* and (x^*, y^*) an *optimal pair* for (P_δ). In particular, any optimal pair satisfies $\|x^*\|_1 = -b^\top y^* - \delta\|y^*\|_1$, i.e., the primal and the dual problem attain the same optimal value. Note that, as a consequence of the optimality conditions (2) and (3), it always holds that $J_P \subseteq J_D$ and $I_D \subseteq I_P$ in case (x^*, y^*) is an optimal pair.

Our approach is to find an optimal pair by repeatedly making use of (2) and (3). Instead of solving (P_δ) directly, we start by setting $\delta^0 \coloneqq \|b\|_\infty$ and

observe that $x^0 = 0$ is an optimal solution of (P_{δ^0}). Now, the main idea behind the iterations of our method is the following: Let $k \in \mathbb{N}_0 \coloneqq \mathbb{N} \cup \{0\}$ and (x^k, y^k) be an optimal pair for (P_{δ^k}). First, we seek a $y^{k+1} \neq y^k$ such that (x^k, y^{k+1}) is still an optimal pair for (P_{δ^k}). After that, we aim at identifying x^{k+1} and $t > 0$ such that with $\delta^{k+1} = \delta^k - t$, (x^{k+1}, y^{k+1}) is an optimal pair for $(\mathrm{P}_{\delta^{k+1}})$. We repeat these steps as long as $\delta^{k+1} > \delta$; when finally $\delta^{k+1} = \delta$, we have found an optimal pair (x^{k+1}, y^{k+1}) for our initial problem (P_δ).

We remark that while (2) and (3) show that, e.g., $y^0 = 0$ would be a valid dual certificate associated with x^0 (other similarly simple choices are possible), such a heuristic choice—then to be used for a first primal update step—may lead to a "zero step" ($t = 0$, $x^1 = x^0$), after which a new dual iterate must be computed. Therefore, in ℓ_1-Houdini, we will actually start with the computation of a dual certificate directly (i.e., we do not need any y^0).

2.2 Primal Updates

Suppose (x^k, y^{k+1}) is an optimal pair for (P_{δ^k}) and we seek x^{k+1} and t such that (x^{k+1}, y^{k+1}) is an optimal pair for $(\mathrm{P}_{\delta^k - t})$. From (2) and (3) we know that x^{k+1} and t must fulfill

$$-A^\top y^{k+1} \in \operatorname{Sign}(x^{k+1}) \quad \text{and} \quad Ax^{k+1} - b \in (\delta^k - t)\operatorname{Sign}(y^{k+1}).$$

The first condition restricts both the support and the sign of x^{k+1}, i.e., it must hold that

$$\begin{aligned} & x_j^{k+1} = 0 && \text{if} \quad |(A^\top y^{k+1})_j| < 1, \\ & x_j^{k+1} \geq 0 && \text{if} \quad (A^\top y^{k+1})_j = -1 \\ \text{and} \quad & x_j^{k+1} \leq 0 && \text{if} \quad (A^\top y^{k+1})_j = 1, \end{aligned}$$

or equivalently,

$$x_{J_D^c}^{k+1} = 0 \quad \text{and} \quad (A_{J_D}^\top y^{k+1}) \odot x_{J_D}^{k+1} \leq 0. \tag{4}$$

We split the second condition and start with the components I_D in which y^{k+1} is non-zero and thus, $\operatorname{Sign}(y_{I_D}^{k+1}) = \operatorname{sign}(y_{I_D}^{k+1})$ is single-valued. This leads us to a linear system in x^{k+1} and t:

$$A^{I_D} x^{k+1} + t \cdot \operatorname{sign}(y_{I_D}^{k+1}) = b_{I_D} + \delta^k \operatorname{sign}(y_{I_D}^{k+1}). \tag{5}$$

The remainder of the second condition dictates the inclusions

$$a_i^\top x^{k+1} - b_i \in [-(\delta^k - t), \delta^k - t] \qquad \text{for all } i \in I_D^c,$$

which are equivalent to the linear constraints

$$-A^{I_D^c} x^{k+1} + t\mathbb{1} \leq \delta^k \mathbb{1} - b_{I_D^c} \qquad \text{and} \qquad A^{I_D^c} x^{k+1} + t\mathbb{1} \leq \delta^k \mathbb{1} + b_{I_D^c}. \tag{6}$$

Finally, intuitive bounds for t are given by

$$0 \leq t \leq \delta^k - \delta. \tag{7}$$

Therein, the lower bound prevents regress and the upper bound ensures that we do not jump over an optimal solution of the original problem (recall that under our assumption $\delta < \|b\|_\infty$, any optimal solution of (P_δ) lies on the boundary of the feasible set).

Note that, by construction, $x^{k+1} = x^k$ and $t = 0$ always yield a solution of (4)–(7). Nevertheless, this choice would imply stagnation (the aforementioned "zero step"). In contrast, we can perform a maximal step with respect to the current iterates (x^k, y^{k+1}) by maximizing t w.r.t. the constraints (4)–(7), which amounts to solving a linear program. (Note that the number of variables is substantially reduced by eliminating $x^{k+1}_{J_D^c}$, which must be zero; typically, J_D will be very small—and hence, J_D^c large—at least in the beginning, although generally this depends on the structure of b.)

In case the maximum objective is $t = 0$, no progress is achievable by performing a primal update; we will see later (cf. Lemma 2.4) that this case, in fact, never occurs during our algorithm. Also, since $t = 0$ is always possible, the lower bound $t \geq 0$ is redundant and can be omitted from (7).

2.3 Dual Updates

The dual update follows the same principle as the primal update except that here, x^k and δ^k are fixed and we seek y^{k+1} such that

$$-A^\top y^{k+1} \in \operatorname{Sign}(x^k) \quad \text{and} \quad Ax^k - b \in \delta^k \operatorname{Sign}(y^{k+1}).$$

Here, the second condition restricts the support and the sign of y^{k+1}, i.e.,

$$y^{k+1}_{I_P^c} = 0 \quad \text{and} \quad -\operatorname{sign}(A^{I_P} x^k - b_{I_P}) \odot y^{k+1}_{I_P} \leq 0. \tag{8}$$

We split the first condition. Starting with the primal support J_P, on which $\operatorname{Sign}(x^k_{J_P}) = \operatorname{sign}(x^k_{J_P})$ is single-valued, we obtain the linear system

$$-A^\top_{J_P} y^{k+1} = \operatorname{sign}(x^k_{J_P}). \tag{9}$$

On the complementary components J_P^c, the first condition yields the linear constraints

$$-\mathbb{1} \leq A^\top_{J_P^c} y^{k+1} \leq \mathbb{1}. \tag{10}$$

Just as in case of the primal update, there is a trivial solution to (8)–(10), namely $y^{k+1} = y^k$. Moreover, we can again exploit that the feasible support I_P of y^{k+1} will, at least in the beginning, be small (so that many variables $y^{k+1}_{I_P^c} = 0$). However, in contrast to the primal update, where it was obvious to maximize t, it is not directly clear which solution we should prefer in case (8)–(10) does not have a unique feasible point. The following theorem of alternatives gives an answer to this problem.

2.4 A Theorem of the Alternative

The following results provide, in particular, a selection rule for the dual update which forms a key element for a working algorithm since it guarantees the subsequent primal update to be successful (i.e., not a "zero step").

The two alternatives (11a)–(11e) and (12a)–(12d) in the lemma below are linear (in-)equality systems that *improvement directions* must obey (when interpreting primal and dual updates as moving from x^k to $x^k + t \cdot d$ and from y^k to $y^k + s \cdot e$, respectively).

Lemma 2.1 *Let $(\hat{x}, \hat{y})$ be an optimal pair for $(P_{\hat{\delta}})$ for some $\delta < \hat{\delta} \leq \|b\|_\infty$. Then, one and only one of the systems*

$$\begin{aligned}
-\operatorname{sign}(A\hat{x} - b)^\top e &< 0 && \text{(11a)}\\
A_{J_P}^\top e &= 0 && \text{(11b)}\\
A_{J_D \setminus J_P}^\top \hat{y} \odot A_{J_D \setminus J_P}^\top e &\leq 0 && \text{(11c)}\\
-\operatorname{sign}(A^{I_P \setminus I_D}\hat{x} - b_{I_P \setminus I_D}) \odot e_{I_P \setminus I_D} &\leq 0 && \text{(11d)}\\
e_{I_P^c} &= 0 && \text{(11e)}
\end{aligned}$$

and

$$\begin{aligned}
A^{I_D} d &= -\operatorname{sign}(\hat{y}_{I_D}) && \text{(12a)}\\
\operatorname{sign}(A^{I_P \setminus I_D}\hat{x} - b_{I_P \setminus I_D}) \odot A^{I_P \setminus I_D} d &\leq -\mathbb{1} && \text{(12b)}\\
A_{J_D \setminus J_P}^\top \hat{y} \odot d_{J_D \setminus J_P} &\leq 0 && \text{(12c)}\\
d_{J_D^c} &= 0. && \text{(12d)}
\end{aligned}$$

has a solution.

Proof With $\Sigma_1 \coloneqq \operatorname{Diag}(\operatorname{sign}(A^{I_P \setminus I_D}\hat{x} - b_{I_P \setminus I_D}))$ and $\Sigma_2 \coloneqq \operatorname{Diag}(A_{J_D \setminus J_P}^\top \hat{y})$, we have $\Sigma_1 = \Sigma_1^{-1}$ as well as $\Sigma_2 = \Sigma_2^{-1}$ and can rewrite the first system as

$$\begin{aligned}
-\mathbb{1}^\top \Sigma_1 e_{I_P \setminus I_D} \quad -\operatorname{sign}(A^{I_D}\hat{x} - b_{I_D})^\top e_{I_D} &< 0\\
-\Sigma_2 (A_{J_D \setminus J_P}^{I_P \setminus I_D})^\top e_{I_P \setminus I_D} \quad -\Sigma_2 (A_{J_D \setminus J_P}^{I_D})^\top e_{I_D} &\geq 0\\
(A_{J_P}^{I_P \setminus I_D})^\top e_{I_P \setminus I_D} \quad +(A_{J_P}^{I_D})^\top e_{I_D} &= 0\\
\Sigma_1 e_{I_P \setminus I_D} &\geq 0.
\end{aligned}$$

We substitute $\hat{e}_{I_P \setminus I_D} \coloneqq \Sigma_1 e_{I_P \setminus I_D}$ and observe that the system has a solution if and only if the system

$$\begin{aligned}
-\mathbb{1}^\top \hat{e}_{I_P \setminus I_D} \quad -\operatorname{sign}(A^{I_D}\hat{x} - b_{I_D})^\top e_{I_D} &< 0\\
-\Sigma_2 (A_{J_D \setminus J_P}^{I_P \setminus I_D})^\top \Sigma_1 \hat{e}_{I_P \setminus I_D} \quad -\Sigma_2 (A_{J_D \setminus J_P}^{I_D})^\top e_{I_D} &\geq 0\\
(A_{J_P}^{I_P \setminus I_D})^\top \Sigma_1 \hat{e}_{I_P \setminus I_D} \quad +(A_{J_P}^{I_D})^\top e_{I_D} &= 0\\
\hat{e}_{I_P \setminus I_D} &\geq 0
\end{aligned}$$

is feasible. By Farkas' Lemma (see, e.g., [26, Corollary 7.1d]), this system has a solution if and only if the associated alternative system

$$\begin{aligned} -\Sigma_1 A^{I_P\setminus I_D}_{J_D\setminus J_P}\Sigma_2\hat{d}_{J_D\setminus J_P} \quad +\Sigma_1 A^{I_P\setminus I_D}_{J_P} d_{J_P} &\leq -\mathbb{1} \\ -A^{I_D}_{J_D\setminus J_P}\Sigma_2\hat{d}_{J_D\setminus J_P} \quad +A^{I_D}_{J_P} d_{J_P} &= -\operatorname{sign}(A^{I_D}\hat{x}-b_{I_D}) \\ \hat{d}_{J_D\setminus J_P} &\geq 0 \end{aligned}$$

is infeasible. Since $\operatorname{sign}(A^{I_D}\hat{x}-b_{I_D}) = \operatorname{sign}(\hat{y}_{I_D})$ and by substituting $d_{J_D\setminus J_P} := -\Sigma_2\hat{d}_{J_D\setminus J_P}$, we obtain that equivalently,

$$\begin{aligned} \Sigma_1 A^{I_P\setminus I_D}_{J_D} d_{J_D} &\leq -\mathbb{1} \\ A^{I_D}_{J_D} d_{J_D} &= -\operatorname{sign}(\hat{y}_{I_D}) \\ -\Sigma_2 d_{J_D\setminus J_P} &\geq 0 \end{aligned}$$

is infeasible. The claim now follows by explicitly including $e_{I_P^c} = 0$ and $d_{J_D^c} = 0$ in the respective systems. □

In fact, our algorithm does not use explicit direction vectors, but the above first set of alternative systems will be useful for the proof of the next result and may also be of interest in its own right.

Theorem 2.2 *Let $(\hat{x},\hat{y})$ be an optimal pair for $(P_{\hat{\delta}})$ for some $\delta < \hat{\delta} \leq \|b\|_\infty$. Then, the following four alternatives are equivalent.*

(I) *The system* (11a)–(11e) *is feasible.*
(II) *The system* (12a)–(12d) *is infeasible.*
(III) *$(\hat{x},0)$ is an optimal solution of*

$$\begin{aligned} \max_{(x,t)\in\mathbb{R}^n\times\mathbb{R}} \quad & t && && && (13a)\\ \text{s.t.} \quad & A^{I_D}x - b_{I_D} &&= (\hat{\delta}-t)\operatorname{sign}(\hat{y}_{I_D}) && && (13b)\\ (t-\hat{\delta})\mathbb{1} \leq\ & A^{I^c_D}x - b_{I^c_D} &&\leq (\hat{\delta}-t)\mathbb{1} && && (13c)\\ & A^\top\hat{y}\odot x &&\leq 0 && && (13d)\\ & x_{J^c_D} &&= 0 && && (13e)\\ & t &&\leq \hat{\delta}-\delta. && && (13f) \end{aligned}$$

(IV) *$\hat{y}$ is not an optimal solution of*

$$\begin{aligned} \min_{y\in\mathbb{R}^m} \quad & -\operatorname{sign}(A\hat{x}-b)^\top y && && && (14a)\\ \text{s.t.} \quad & -A^\top_{J_P} y &&= \operatorname{sign}(\hat{x}_{J_P}) && && (14b)\\ -\mathbb{1} \leq\ & -A^\top_{J^c_P} y &&\leq \mathbb{1} && && (14c)\\ & -\operatorname{sign}(A\hat{x}-b)\odot y &&\leq 0 && && (14d)\\ & y_{I^c_P} &&= 0. && && (14e) \end{aligned}$$

Proof Lemma 2.1 already shows that alternatives (I) and (II) are equivalent.

Moreover, since $(\hat{x}, \hat{y})$ forms an optimal pair for $(\mathrm{P}_{\hat{\delta}})$, several relations corresponding to constraints in the optimization problems of alternatives (III) and (IV) already hold true, by the optimality conditions and the definitions of the respective index sets: Due to (2), (14b) and (14c) are satisfied, and due to (3), so are (13b) and (13c) for $t = 0$, i.e., we have

$$\begin{array}{rcrcrcl} -A_{J_P}^\top \hat{y} & = & \operatorname{sign}(\hat{x}_{J_P}), & -\mathbb{1} \leq & -A_{J_P^c}^\top \hat{y} & \leq & \mathbb{1}, \\ A^{I_D}\hat{x} - b_{I_D} & = & \hat{\delta}\operatorname{sign}(\hat{y}_{I_D}), & -\hat{\delta}\mathbb{1} \leq & A^{I_D^c}\hat{x} - b_{I_D^c} & \leq & \hat{\delta}\mathbb{1}. \end{array}$$

By definition of the active sets I_P and J_D together with (2) and (3) (in other words, by complementary slackness) , (13e) and (14e) are also satisfied, i.e., $\hat{x}_{J_D^c} = 0$ and $\hat{y}_{I_P^c} = 0$. Finally, (13d) follows from (2) and (14d) from (3), and since $\hat{x}_j \neq 0$ for all $j \in J_P$ and $\hat{y}_i \neq 0$ for all $i \in I_D$, we obtain, in particular, that

$$\begin{aligned} A_{J_P}^\top \hat{y} \odot \hat{x}_{J_P} &< 0 \\ \text{and} \qquad -\operatorname{sign}(A^{I_D}\hat{x} - b_{I_D}) \odot \hat{y}_{I_D} &< 0. \end{aligned}$$

Keeping the above relations in mind, we proceed to show the equivalence of alternatives (II) and (III):

Suppose that alternative (II) is *not* true, i.e., there exists a d that satisfies (12a)–(12d). As d fulfills (12a) and (12d), we get that for each $t > 0$, $(\hat{x}+td, t)$ fulfills (13b) and (13e), respectively. From (12b) we obtain the existence of a $t_1 > 0$ such that $(\hat{x} + td, t)$ satisfies (13c) for all $0 \leq t \leq t_1$, and because of (12c), there exists a $t_2 > 0$ such that $(\hat{x} + td, t)$ fulfills (13d) for all $0 \leq t \leq t_2$. Consequently, we can choose $t = \min(t_1, t_2, \hat{\delta}-\delta) > 0$ and have a corresponding feasible solution $(\hat{x} + td, t)$ of (12a)–(12d), which shows that alternative (III) is not true either.

Conversely, suppose that alternative (III) is not true, i.e., there exists a pair (x, t) with $t > 0$ that satisfies (13b)–(13f). We easily see that $d = (x - \hat{x})/t$ obeys (12a). Obviously, by construction, also (12d) holds for d. Moreover, it holds that

$$\begin{aligned} &\operatorname{sign}(A^{I_P\setminus I_D}\hat{x} - b_{I_P\setminus I_D}) \odot A^{I_P\setminus I_D} d \\ =&\tfrac{1}{t}\operatorname{sign}(A^{I_P\setminus I_D}\hat{x} - b_{I_P\setminus I_D}) \odot ([A^{I_P\setminus I_D}x - b_{I_P\setminus I_D}] - [A^{I_P\setminus I_D}\hat{x} - b_{I_P\setminus I_D}]) \\ =&\tfrac{1}{t}\operatorname{sign}(A^{I_P\setminus I_D}\hat{x} - b_{I_P\setminus I_D}) \odot \operatorname{sign}(A^{I_P\setminus I_D}x - b_{I_P\setminus I_D}) \odot |A^{I_P\setminus I_D}x - b_{I_P\setminus I_D}| \\ &-\tfrac{1}{t}\operatorname{sign}(A^{I_P\setminus I_D}\hat{x} - b_{I_P\setminus I_D}) \odot \operatorname{sign}(A^{I_P\setminus I_D}\hat{x} - b_{I_P\setminus I_D}) \odot |A^{I_P\setminus I_D}\hat{x} - b_{I_P\setminus I_D}| \\ \leq&\tfrac{\hat{\delta}-t}{t}\mathbb{1} - \tfrac{\hat{\delta}}{t}\mathbb{1} = -\mathbb{1}, \end{aligned}$$

so d satisfies (12b) as well. Finally, (12c) also holds true, since

$$A_{J_D\setminus J_P}^\top \hat{y} \odot d_{J_D\setminus J_P} = \tfrac{1}{t}\underbrace{A_{J_D\setminus J_P}^\top \hat{y} \odot x_{J_D\setminus J_P}}_{\leq 0} - \tfrac{1}{t}A_{J_D\setminus J_P}^\top \hat{y} \odot \underbrace{\hat{x}_{J_D\setminus J_P}}_{=0} \leq 0.$$

Thus, we conclude that alternative (II) is indeed not true either.

To complete the proof, it now suffices to show that alternatives (I) and (IV) are equivalent. First, suppose that alternative (I) is true, i.e., there exists an e that satisfies (11a)–(11e). For arbitrary $s > 0$, the vector $\hat{y}+se$ still obeys (14b) and (14e), because of (11b) and (11e), respectively. Furthermore, (11c) ensures that there exists an $s_1 > 0$ such that $\hat{y} + se$ still satisfies (14c) for $0 \leq s \leq s_1$, and (11d) ensures the existence of an $s_2 > 0$ such that $\hat{y} + se$ obeys (14d) for $0 \leq s \leq s_2$. Thus, we can choose $s = \min(s_1, s_2)$ and obtain that $\hat{y} + se$ satisfies (14b)–(14e). Moreover, (11a) shows that $-\operatorname{sign}(A\hat{x} - b)^\top(\hat{y} + se) < -\operatorname{sign}(A\hat{x} - b)^\top\hat{y}$ and it follows that $\hat{y}$ is *not* the minimizer of (14a)–(14e) and thus, that alternative (IV) is true.

Now, suppose conversely that alternative (IV) is true and that $y \neq \hat{y}$ is a minimizer of (14a)–(14e). Then, $e \coloneqq y - \hat{y}$ satisfies $-\operatorname{sign}(A\hat{x} - b)^\top(\hat{y} + e) < -\operatorname{sign}(A\hat{x} - b)^\top\hat{y}$, which shows that e obeys (11a). Moreover, (14b)–(14e) continue to hold for $\hat{y} + e$, which implies that e satisfies (11b)–(11e) as well, and consequently, that alternative (I) is true. □

In (13b)–(13f) and (14b)–(14e), we recognize the primal and dual update conditions derived in the previous two subsections. As this connection will be essential for our algorithm, we formalize it in the following lemma:

Lemma 2.3 *Let $(\hat{x}, \hat{y})$ be an optimal pair for $(P_{\hat{\delta}})$ for some $\delta < \hat{\delta} \leq \|b\|_\infty$. Then, $(\hat{x}, \tilde{y})$ is an optimal pair for $(P_{\hat{\delta}})$ if and only if $\tilde{y}$ is feasible for* (14b)–(14e). *Vice versa, $(\tilde{x}, \hat{y})$ is an optimal pair for $(P_{\hat{\delta}-\tilde{t}})$ with $\delta \leq \hat{\delta} - \tilde{t}$ if and only if $(\tilde{x}, \tilde{t})$ is feasible for* (13b)–(13f).

Proof The first statement holds because the conditions (14b)–(14e) are equivalent to (8)–(10) with $\hat{x} = x^k$ and $y = y^{k+1}$.The second statement follows because the conditons (13b)–(13f) are equivalent to (4)–(7) with $\hat{\delta} = \delta^k$, $\hat{y} = y^{k+1}$, $x = x^{k+1}$ and $t = t^{k+1}$.

2.5 ℓ_1-Houdini Algorithm and Finite Termination

Theorem 2.2 and Lemma 2.3 suggest the following algorithm: For a given $\delta^k > \delta$ and an optimal pair (x^k, y^k) do: First update y^{k+1} as a solution to (14a)–(14e) (with $\hat{x} = x^k$) and then find an updated x^{k+1} and a $t^{k+1} > 0$ as solution of (13a)–(13f) (with $\hat{y} = y^{k+1}$ and $\hat{\delta} = \delta^k$). In detail this is described in Algorithm 1.

To prove convergence of Algorithm 1 we start with a lemma:

Lemma 2.4 *In each two consecutive iterations, Algorithm 1 produces iterates $y^{k+1} \neq y^k$ and $x^{k+1} \neq x^k$. In particular, it holds that $t^{k+1} > 0$ in each iteration and if (x^k, y^k) is an optimal pair for (P_{δ^k}), then (x^{k+1}, y^{k+1}) is an optimal pair for $(P_{\delta^{k+1}})$ with $\delta^{k+1} < \delta^k$.*

Proof In the beginning, we have $x^0 = 0$ and determine y^1 solving (14a)–(14e) with $\hat{x} = x^0$. By Lemma 2.3, (x^0, y^1) is an optimal pair for (P_{δ^0}). By

Algorithm 1: ℓ_1-HOUDINI

Input: $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, $0 \leq \delta < \|b\|_\infty$
Output: solution x^* to problem (P_δ)

// Initialization:
$\delta^0 \leftarrow \|b\|_\infty$
$x^0 \leftarrow 0$
$I_P \leftarrow \{i : |b_i| = \delta^0\}$
$J_P \leftarrow \emptyset$
$k \leftarrow 0$
repeat
  // Dual update:
  $y^{k+1} \leftarrow$ solution of problem (14a)–(14e) with $\hat{x} = x^k$
  $I_D \leftarrow \{i : y_i^{k+1} \neq 0\}$
  $J_D \leftarrow \{j : |A_j^\top y^{k+1}| = 1\}$
  // Primal update:
  $(x^{k+1}, t^{k+1}) \leftarrow$ sol. of problem (13a)–(13f) with $\hat{y} = y^{k+1}$ and $\hat{\delta} = \delta^k$
  $\delta^{k+1} \leftarrow \delta^k - t^{k+1}$
  $I_P \leftarrow \{i : |a_i^\top x^{k+1} - b_i| = \delta^{k+1}\}$
  $J_P \leftarrow \{j : x_j^{k+1} \neq 0\}$
  $k \leftarrow k + 1$
until $\delta^k = \delta$
return $x^* = x^{k+1}$

Theorem 2.2, $(x^0, 0)$ is not an optimal solution to (13a)–(13f) with $\hat{y} = y^1$ and $\hat{\delta} = \delta^0$. It follows that $x^1 \neq x^0$ and $t^1 > 0$ after solving (13a)–(13f). By Lemma 2.3, it further holds that (x^1, y^1) is an optimal pair for (P_{δ^1}) with $\delta^1 = \delta^0 - t^1 < \delta^0$.

Now suppose $k \geq 1$ and consider an iteration of Algorithm 1 starting from an optimal pair (x^k, y^k) for (P_{δ^k}) which is known from the previous iteration. First, we determine a new dual iterate y^{k+1} by solving (14a)–(14e) with $\hat{x} = x^k$. From the previous primal update we know that $(x^k, \delta^{k-1} - \delta^k)$ is a solution of (13a)–(13f) with $\hat{\delta} = \delta^{k-1}$ and $\hat{y} = y^k$. It follows that $(x^k, 0)$ is a solution of (13a)–(13f) with $\hat{\delta} = \delta^k$ and $\hat{y} = y^k$. In turn, Theorem 2.2 states that y^k is not a solution of (14a)–(14e) with $\hat{x} = x^k$. By construction, y^{k+1} is a solution of (14a)–(14e) with $\hat{x} = x^k$ and consequently $y^{k+1} \neq y^k$. Moreover, it follows from Lemma 2.3 that (x^k, y^{k+1}) is an optimal pair for (P_{δ^k}). Vice versa, Theorem 2.2 states that $(x^k, 0)$ is (although feasible) not a solution of (13a)–(13f) with $\hat{y} = y^{k+1}$ and $\hat{\delta} = \delta^k$. Since (x^{k+1}, t^{k+1}) is exactly such a solution, it follows that $t^{k+1} > 0$ and $x^{k+1} \neq x^k$ and again by Lemma 2.3, (x^{k+1}, y^{k+1}) is an optimal pair for $(\mathrm{P}_{\delta^{k+1}})$ with $\delta^{k+1} = \delta^k - t^{k+1} < \delta^k$. □

Certainly, Lemma 2.4 does not yet prove convergence of Algorithm 1. Nevertheless, we see that each iteration contributes at least a small approach towards a solution of (P_δ).

Theorem 2.5 *Algorithm 1 terminates after a finite number of iterations and returns an optimal solution of (P_δ).*

Proof The number of possible support sets J_P, active sets I_P, associated sign patterns and combinations thereof is finite. Suppose that for $k < \ell$ Algorithm 1 produces $J_P := J_P^k = J_P^\ell$, $I_P := I_P^k = I_P^\ell$, $\operatorname{sign}(x_{J_P}^k) = \operatorname{sign}(x_{J_P}^\ell)$ and $\operatorname{sign}(Ax^k - b) = \operatorname{sign}(Ax^\ell - b)$. According to (14a)–(14e) we obtain that also $y^{k+1} = y^{\ell+1}$. It follows that the primal update steps (13a)–(13f) to find x^{k+1} and $x^{\ell+1}$ are equal except that we have $\hat{\delta} = \delta^k$ in the first case and $\hat{\delta} = \delta^\ell$ in the second, where $\delta^k > \delta^\ell$ by Lemma 2.4. Since $\hat{\delta}$ is a constant, it is equivalent to rewrite (13a) as $t - \hat{\delta}$. The substitution $\tilde{\delta} := \hat{\delta} - t$ in (13a)–(13c) and (13f) then reveals that the update problems for x^{k+1} and $x^{\ell+1}$ indeed have an identical reformulation. Hence, we obtain the same optimal value for $\tilde{\delta}$ in both cases, which shows that $\delta^{k+1} = \delta^{\ell+1}$ and contradicts Lemma 2.4 since $k < \ell$. Thus, Algorithm 1 terminates after a finite number of iterations with an optimal solution. □

Remark 2.6 As a direct consequence of the preceding proof, we conclude that the number of iterations needed by Algorithm 1 to find an optimal solution of (P_δ) does not exceed 3^{m+n}. This corresponds to the number of possible combinations of primal support sets, primal active sets and associated sign patterns. In fact, using similar arguments as in [19] (opposing sign patterns cannot occur along the solution path), we can even show that the number of iterations is at most $(3^{m+n} + 1)/2$.

3 Practical Considerations

As mentioned earlier, one may in principle use an arbitrary LP solver to tackle the update problems in ℓ_1-Houdini. However, due to their special structure, we found active-set strategies to be particularly efficient for these LPs. In the following, we give the details of our approach; the numerical experiments in Section 4 will later demonstrate the efficiency of our corresponding implementation.

3.1 Active-Set Method for the Primal Update

Finding a new primal iterate x^{k+1} and the related decrease t^{k+1} of the homotopy parameter in Step 10 of Algorithm 1 gives rise to the linear program

$$\max_{(x_{J_D},t)\in\mathbb{R}^{|J_D|}\times\mathbb{R}} \quad \begin{pmatrix}0\\1\end{pmatrix}^\top \begin{pmatrix}x_{J_D}\\t\end{pmatrix} \tag{15a}$$

$$\text{s.t.} \quad \left[A^{I_D}_{J_D}\ \operatorname{sign}(y^{k+1}_{I_D})\right] \begin{pmatrix}x_{J_D}\\t\end{pmatrix} = \delta^k \operatorname{sign}(y^{k+1}_{I_D}) + b_{I_D} \tag{15b}$$

$$\begin{bmatrix} A^{I^c_D}_{J_D} & \mathbb{1} \\ -A^{I^c_D}_{J_D} & \mathbb{1} \\ 0 & 1 \end{bmatrix} \begin{pmatrix}x_{J_D}\\t\end{pmatrix} \leq \begin{pmatrix} \delta^k\mathbb{1} + b_{I^c_D} \\ \delta^k\mathbb{1} - b_{I^c_D} \\ \delta^k - \delta \end{pmatrix} \tag{15c}$$

$$\begin{bmatrix} \operatorname{Diag}(A^\top_{J_D} y^{k+1}) & 0 \\ 0 & -1 \end{bmatrix} \begin{pmatrix}x_{J_D}\\t\end{pmatrix} \leq 0. \tag{15d}$$

In this section, we introduce an active-set method in order to solve problem (15a)-(15d). The idea for our approach is based upon the active-set method for quadratic programs illustrated, e.g., in [21]. We adapt the method to the special type of linear programs that we are faced with. We refer to Appendix A for the general procedure and to Table 5 for supplementary details about the implementation of the algorithm.

3.1.1 Initialization

We observe that the point $(x^k_{J_D}, 0)$ is feasible since (x^k, y^{k+1}) is an optimal pair for (P_{δ^k}). We set $\ell = 0$ and choose our starting point $(\xi^\ell_{J_D}, \tau^\ell) = (x^k_{J_D}, 0)$ accordingly. Regarding (15c), we see that the subset of active constraints at the starting point $(x^k_{J_D}, 0)$ corresponds to $\mathcal{A} = I_P \setminus I_D$ with either positive or negative sign. The initial support is exactly $\mathcal{S} = J_P$.

The variable t represents the decrease of the homotopy parameter starting from δ^k. Although the associated iterate is initially zero, t joins the support once we have performed a step towards an ascent direction. Since each constructed direction is an ascent direction, t does not leave the support afterwards. Consequently, we have $\mathcal{S} = J_P \cup \{t\}$ and $t \geq 0$ throughout, and the constraint $-t \leq 0$ in (15d) can actually be omitted. We only keep it in order to adapt (15) to problem (41) (see Appendix A below) from which we derive our active-set method.

The constraint $t \leq \delta^k - \delta$ is neither active in the beginning nor will it be so unless we have found an optimal solution of our original problem (P_δ).

3.1.2 Ascent Directions and Blocking Constraints

To find an ascent direction (d, d_t) preserving $\mathcal{A}$ and $\mathcal{S}$, we fix $d_{J_D \setminus J_P} = 0$ and $d_t = 1$ and seek a solution of the linear system

$$A^{I_P}_{J_P} d_{J_P} = -\operatorname{sign}(A^{I_P}\xi^\ell - b_{I_P}). \tag{16}$$

If a solution of (16) exists, the largest step size α preserving feasibility is

$$\alpha = \min\left\{\alpha_{\mathcal{A}},\, \alpha_{\mathcal{S}},\, \delta^k - \tau^\ell - \delta\right\}, \tag{17}$$

wherein

$$\alpha_{\mathcal{A}} = \min\left\{ \min_{\substack{i\in I_P^c \\ a_i^\top d > -1}} \frac{\delta^k - \tau^\ell - a_i^\top \xi^\ell + b_i}{a_i^\top d + 1},\ \min_{\substack{i\in I_P^c \\ a_i^\top d < 1}} \frac{\delta^k - \tau^\ell + a_i^\top \xi^\ell - b_i}{-a_i^\top d + 1} \right\} \tag{18}$$

and

$$\alpha_{\mathcal{S}} = \min_{\substack{j\in J_P \\ A_j^\top y^{k+1}\cdot d_j > 0}} -\frac{\xi_j^\ell}{d_j}. \tag{19}$$

The new iterates are then

$$\xi^{\ell+1} = \xi^\ell + \alpha d \qquad \text{and} \qquad \tau^{\ell+1} = \tau^\ell + \alpha. \tag{20}$$

In case $\alpha = \delta^k - \tau^\ell - \delta$, we stop thereafter since $x^* = \xi^{\ell+1}$ is an optimal solution of (P_δ). Otherwise, we finally update

$$\begin{aligned} I_P &= I_P \cup \left\{ i \in I_P^c : |A^i \xi^{\ell+1} - b_i| = \delta^k - \tau^{\ell+1} \right\} \\ J_P &= J_P \setminus \left\{ j \in J_P : |\xi^{\ell+1}| = 0 \right\} \end{aligned} \tag{21}$$

which corresponds to an update of $\mathcal{A} = I_P \setminus I_D$ and $\mathcal{S} = J_P \cup \{t\}$.

3.1.3 Lagrange Multipliers

If a solution of (16) does not exist, zero is an optimal solution of

$$\max_{(d_{J_P}, d_t)\in\mathbb{R}^{|J_P|}\times\mathbb{R}} \begin{pmatrix} 0 \\ 1 \end{pmatrix}^\top \begin{pmatrix} d_{J_P} \\ d_t \end{pmatrix} \quad \text{s.t.} \quad \left[A_{J_P}^{I_P}\ \operatorname{sign}(A^{I_P}\xi^\ell - b_{I_P})\right] \begin{pmatrix} d_{J_P} \\ d_t \end{pmatrix} = 0$$

and the associated KKT conditions show that there exists $\hat{e}_{I_P}$ satisfying

$$\begin{aligned} (A_{J_P}^{I_P})^\top \hat{e}_{I_P} &= 0 \\ \operatorname{sign}(A^{I_P}\xi^\ell - b_{I_P})^\top \hat{e}_{I_P} &= 1. \end{aligned} \tag{22}$$

Building on that, we set

$$\mu_{I_P\setminus I_D} = \operatorname{sign}(A^{I_P\setminus I_D}\xi^\ell - b_{I_P\setminus I_D}) \odot \hat{e}_{I_P\setminus I_D} \tag{23}$$

$$\nu_{J_D\setminus J_P} = -(A_{J_D\setminus J_P}^\top y^{k+1}) \odot (A_{J_D\setminus J_P}^{I_P})^\top \hat{e}_{I_P}, \tag{24}$$

where the components of the vector $A_{J_D\setminus J_P}^\top y^{k+1}$ are all ± 1.

We can consider $\mu_{I_P\setminus I_D}$ and $\nu_{J_D\setminus J_P}$ as Lagrange multipliers associated with the KKT conditions for (15). In particular, $\mu_{I_P\setminus I_D}$ corresponds to the set $\mathcal{A}$ of active constraints in (15c) and $\nu_{J_D\setminus J_P}$ to $\mathcal{S}^c$, i.e., the active constraints in (15d). Although differently motivated, the multipliers (23) and (24) are exactly what we get if we determine $\mu_{\mathcal{A}}$ and $\nu_{\mathcal{S}^c}$ according to Appendix A.4.

In case $\mu_{I_P\setminus I_D} \geq 0$ and $\nu_{J_D\setminus J_P} \geq 0$, the current iterate $\xi_{J_D}^\ell$ is optimal. Else, we pick $i \in I_P \setminus I_D$ with $\mu_i < 0$ or $j \in J_D \setminus J_P$ with $\nu_j < 0$ and update $I_P = I_P \setminus \{i\}$ or $J_P = J_P \cup \{j\}$, respectively. This corresponds to an update of $\mathcal{A}$ and $\mathcal{S}$, respectively.

3.2 Active-Set Method for the Dual Update

Finding a new dual iterate y^{k+1} in Step 7 of Algorithm 1 gives rise to to the linear program

$$\begin{aligned}
\min_{y_{I_P} \in \mathbb{R}^{|I_P|}} \quad & -\operatorname{sign}(A^{I_P} x^k - b_{I_P})^\top y_{I_P} && \text{(25a)}\\
\text{s.t.} \quad & (-A^{I_P}_{J_P})^\top y_{I_P} = \operatorname{sign}(x^k_{J_P}) && \text{(25b)}\\
& \begin{bmatrix} (A^{I_P}_{J_P^c})^\top \\ (-A^{I_P}_{J_P^c})^\top \end{bmatrix} y_{I_P} \geq -\mathbb{1} && \text{(25c)}\\
& \operatorname{Diag}(\operatorname{sign}(A^{I_P} x^k - b_{I_P})) y_{I_P} \geq 0. && \text{(25d)}
\end{aligned}$$

Analogous to the primal case, we use our results from Appendix A to develop an active-set method for problem (25a)–(25d). We refer to Table 4 for additional information on the implementation of the algorithm.

3.2.1 Initialization

In the beginning, $y^k_{I_P}$ is feasible since (x^k, y^k) is an optimal pair. We set $\ell = 0$ and choose $\psi^\ell_{I_P} = y^k_{I_P}$ as our starting point. In view of (25c), the set of active constraints at $y^k_{I_P}$ corresponds to $\mathcal{A} = J_D \setminus J_P$ with either positive or negative sign and the initial support is $\mathcal{S} = I_D$.

3.2.2 Descent Direction and Blocking Constraints

We seek for a descent direction preserving $\mathcal{A}$ and $\mathcal{S}$ by solving

$$\begin{aligned}
(A^{I_D}_{J_D})^\top e_{I_D} &= 0\\
\operatorname{sign}(A^{I_D} x^k - b_{I_D})^\top e_{I_D} &= 1.
\end{aligned} \tag{26}$$

If such a direction exists, the largest step size preserving feasibility is

$$\alpha = \min\left\{\alpha_{\mathcal{A}}, \alpha_{\mathcal{S}}\right\}. \tag{27}$$

Here,

$$\alpha_{\mathcal{A}} = \min\left\{ \min_{\substack{j \in J_D^c \\ A_j^\top e < 0}} \frac{1 + A_j^\top \psi^\ell}{-A_j^\top e}, \min_{\substack{j \in J_D^c \\ A_j^\top e > 0}} \frac{1 - A_j^\top \psi^\ell}{A_j^\top e} \right\} \tag{28}$$

and

$$\alpha_{\mathcal{S}} = \min_{\substack{i \in I_D \\ \operatorname{sign}(a_i^\top x^k - b_i) e_i < 0}} -\frac{\psi^\ell_i}{e_i}. \tag{29}$$

The new iterate is $\psi^{\ell+1} = \psi^\ell + \alpha e$. Finally, we need to update

$$\begin{aligned}
I_D &= I_D \setminus \left\{ i \in I_D : \psi^{\ell+1} = 0 \right\}\\
J_D &= J_D \cup \left\{ j \in J_D^c : |A_j^\top \psi^{\ell+1}| = 1 \right\}
\end{aligned} \tag{30}$$

which corresponds to an upate of $\mathcal{A} = J_D \setminus J_P$ and $\mathcal{S} = I_D$.

3.2.3 Lagrange Multipliers

If a solution of (26) does not exist, then zero is an optimal solution of

$$\min_{e_{I_D} \in \mathbb{R}^{|I_D|}} \; -\operatorname{sign}(A^{I_D} x^k - b_{I_D})^\top e_{I_D} \quad \text{s.t. } (A^{I_D}_{J_D})^\top e_{I_D} = 0.$$

Analogous to above, KKT conditions ensure that there exists $\hat{d}_{J_D}$ such that

$$A^{I_D}_{J_D} \hat{d}_{J_D} = -\operatorname{sign}(A^{I_D} x^k - b_{I_D}) \tag{31}$$

and we obtain Lagrange multipliers for (25) by setting

$$\mu_{J_D \setminus J_P} = -(A^\top_{J_D \setminus J_P} \psi^\ell) \odot \hat{d}_{J_D \setminus J_P} \tag{32}$$

$$\nu_{I_P \setminus I_D} = -\operatorname{sign}(A^{I_P \setminus I_D} x^k - b_{I_P \setminus I_D}) \odot A^{I_P \setminus I_D}_{J_D} \hat{d}_{J_D} - \mathbb{1}, \tag{33}$$

where the components of the vectors $A^\top_{J_D \setminus J_P} \psi^\ell$ and $A^{I_P \setminus I_D}_{J_D} \hat{d}_{J_D}$ are all ± 1.

Here, $\mu_{J_D \setminus J_P}$ corresponds to the set $\mathcal{A}$ of active constraints in (25c) and $\nu_{I_P \setminus I_D}$ correpsonds to $\mathcal{S}^c$, i.e., the set of active constraints in (25d). These multipliers are equal to those we obtain according to Appendix A.4.

In case $\mu_{J_D \setminus J_P} \geq 0$ and $\nu_{I_P \setminus I_D} \geq 0$, the current iterate $\psi^\ell_{I_P}$ is optimal. Otherwise, we can find $j \in J_D \setminus J_P$ with $\mu_j < 0$ or $i \in I_P \setminus I_D$ with $\nu_i < 0$ and update $J_D = J_D \setminus \{j\}$ or $I_D = I_D \setminus \{i\}$, repsectively.

3.3 Links Between Primal and Dual Active-Set Method

In the following, we establish a close connection between the methods discussed in Sections 3.1 and 3.2. This natural link will turn out to be enormously useful in terms of computational efficiency.

In the context of Section 3.1.3, suppose that we have found $\hat{e}_{I_P}$ satisfying equations (22) such that the associated Lagrange multipliers $\mu_{I_P \setminus I_D}$ and $\nu_{J_D \setminus J_P}$ are throughout non-negative. In that situation, we have found an optimal solution of the primal subproblem (15) and proceed to the dual subproblem (25). Therein, we would first attempt to find a direction e_{I_D} satisfying (26). Can this ever be successful?

Let us recall the situation at the end of the previous dual update. In fact, we did not find a direction satisfying (26) and afterwards found that our current iterate was already optimal. Since then, the sets I_D and J_D did not change. Hence, it would be pointless to search a solution of (26) as a first step of the active-set method for the dual update.

As we have argued so far, we would continue by adapting the sets I_D and J_D invoking Lagrange multipliers according to (31)–(33). But there is a

remedy. A comparison of what we have and what we seek for, $\hat{e}_{I_P}$ and e_{I_D}, respectively, reveals the following:

$$\begin{aligned} (A^{I_P}_{J_P})^\top \hat{e}_{I_P} &= 0 & (A^{I_D}_{J_D})^\top e_{I_D} &= 0 \\ \operatorname{sign}(A^{I_P} x^k - b_{I_P})^\top \hat{e}_{I_P} &= 1 & \operatorname{sign}(A^{I_D} x^k - b_{I_D})^\top e_{I_D} &= 1. \end{aligned}$$

The crucial idea is now to perform the updates

$$\begin{aligned} I_D &= I_D \cup \{i \in I_P \setminus I_D : \hat{e}_i \neq 0\} \\ J_D &= J_D \setminus \{j \in J_D \setminus J_P : (A^{I_P}_j)^\top \hat{e}_{I_P} \neq 0\}. \end{aligned} \tag{34}$$

After that, $e_{I_D} = \hat{e}_{I_D}$ will do exactly what we need.

The fact that the Lagrange multipliers associated with $\hat{e}_{I_P}$ are non-negative throughout shows that a non-trivial step $y^k + \alpha\hat{e}$ maintains primal-dual optimality. For $i \in I_P \setminus I_D$ with $\hat{e}_i \neq 0$, it holds that $\operatorname{sign}(a_i^\top x^k - b_i)\hat{e}_i > 0$, which shows that a step in direction $\hat{e}$ provides the dual variable with the desired sign. Further, it holds for $j \in J_D \setminus J_P$ with $A_j^\top \hat{e} \neq 0$ that $A_j^\top y^k \cdot A_j^\top \hat{e} < 0$, which shows that a step in direction $\hat{e}$ forces the respective dual constraint to become inactive while maintaining feasibility.

It is not at all surprising that an analogous approach works in the beginning of the primal update. Suppose that we have $\hat{d}_{J_D}$ according to (31) at hand and the associated Lagrange multipliers are non-negative. We compare $\hat{d}_{J_D}$ to the sought after direction d_{J_P}:

$$A^{I_D}_{J_D} \hat{d}_{J_D} = -\operatorname{sign}(A^{I_D} x^k - b_{I_D}) \qquad A^{I_P}_{J_P} d_{J_P} = -\operatorname{sign}(A^{I_P} x^k - b_{I_P}).$$

Analogous to above, we perform the update

$$\begin{aligned} J_P &= J_P \cup \{j \in J_D \setminus J_P : \hat{d}_j \neq 0\} \\ I_P &= I_P \setminus \{i \in I_P \setminus I_D : a_i^\top \hat{d} \neq -\operatorname{sign}(a_i^\top x^k - b_i)\}, \end{aligned} \tag{35}$$

whereafter $d_{J_P} = \hat{d}_{J_P}$ does the job.

By non-negativity of the Lagrange multipliers associated with $\hat{d}_{J_D}$, it can be shown that a non-trivial step $x^k + \alpha\hat{d}$ maintains primal-dual optimality: For $j \in J_D \setminus J_P$ with $\hat{d}_j \neq 0$ it holds that $-A_j^\top y^{k+1} \cdot \hat{d}_j > 0$. Further, each $i \in I_P \setminus I_D$ with $a_i^\top \hat{d} \neq -\operatorname{sign}(a_i^\top x^k - b_i)$ satisfies $a_i^\top \hat{d} \cdot \operatorname{sign}(a_i^\top x^k - b_i) < -1$.

4 Applications and Examples

In this section, we illustrate the applicability of ℓ_1-HOUDINI with several examples.

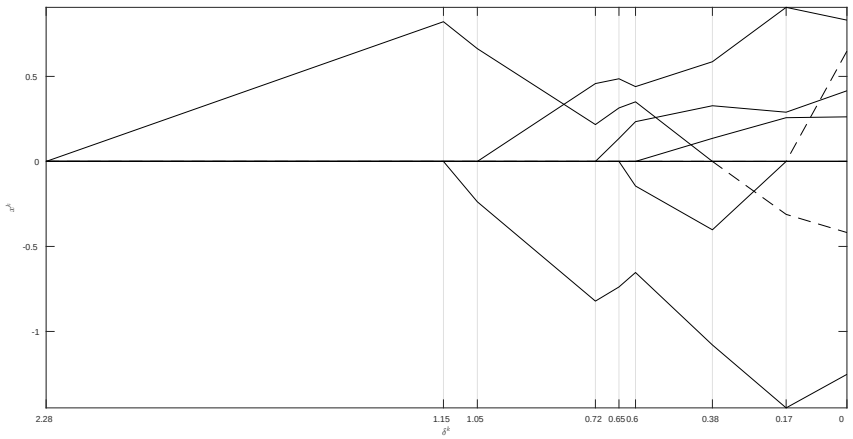

Fig. 1 Examplary run of ℓ_1-Houdini (using active set) with $A \in \mathbb{R}^{6\times 12}$ and $b \in \mathbb{R}^6$ randomly generated and $\delta = 0$. The algorithm needed 9 iterations to solve the problem. Horizontal labels display the value of the homotopy parameter δ^k after each iteration. The plots represent the solution paths of x_j^k for $j = 1, \ldots, 12$. The optimal solution has 6 nonzero entries. The transitions from solid to dashed lines indicate that one variable leaves the support and a different variable enters the support at the respective points.

Example 4.1 (Solution path for a small problem) A typical run of ℓ_1-Houdini on a small problem is shown in Figure 1. We observe that the solution path does not need to show any particular monotonicity; other examples exhibit even more tangled solution paths with multiple variables entering or leaving the support or dense clusters of break points of δ^k at various values.

Example 4.2 (Cross-validation for Chebyshev estimation) The problem (P_δ) is also an instance of linear Chebychev estimation [1,28] (also called method of least maximum absolute deviation [27, Section 2.5]) under a sparsity assumption: We consider samples $\{(b_i, a_i)\}_{i=1}^m \subseteq \mathbb{R} \times \mathbb{R}^n$ and an associated linear model $b = Ax + \eta$, where the dependent variables are aggregated in $b = (b_1, \ldots, b_m)^\top \in \mathbb{R}^m$, the rows $a_i^\top$ of $A \in \mathbb{R}^{m\times n}$ are the independent variables, $\eta \in \mathbb{R}^m$ is a random error term and $x \in \mathbb{R}^n$ is the sought linear predictor. Additionally, let us assume that the errors η_i are i.i.d. and uniformly distributed on an interval $[-\delta, \delta]$, where the *distribution parameter* $\delta > 0$ *is unknown*, and that the *linear predictor* x *is sparse*. If the parameter δ was known a priori, then our assumptions would give rise to the estimate

$$\hat{x} \coloneqq \operatorname*{argmin}_{x\in\mathbb{R}^n} \|x\|_1 \quad \text{s.t. } \|Ax - b\|_\infty \leq \delta \tag{36}$$

which is a particular instance of (P_δ). However, in case δ is unknown, a widely used technique to select a parameter $\hat{\delta}$ associated with an accurate predictive model is cross-validation [17]. In the following example, we demonstrate that the availability of the entire solution path of (36) can be particularly useful in this context.

We use a K-fold cross-validation scheme where $S \coloneqq \{(b_i, a_i)\}_{i=1}^m$ is randomly subdivided into equally sized folds $S_1, \ldots, S_K$. Accordingly, we define index sets $I_k \coloneqq \{i : (b_i, a_i) \in S_k\}$. For $k = 1, \ldots, K$, we calculate solution paths $x_k(\delta) : [\delta_{\min}, \infty) \to \mathbb{R}^n$ by applying ℓ_1-HOUDINI to the problems

$$\min_{x \in \mathbb{R}^n} \ \|x\|_1 \quad \text{s.t. } \|A^{I_k^c} x - b_{I_k^c}\|_\infty \leq 0. \tag{P_k}$$

Note that (P_k) does not necessarily have a feasible solution w.r.t. the parameter $\delta = 0$. However, ℓ_1-HOUDINI can easily be modified (stop as soon as the step size is zero) in order to return solution paths $x_k(\delta) : [\delta_{\min}^k, \infty) \to \mathbb{R}^n$, where $\delta_{\min}^k$ is the smallest parameter such that (P_k) has a feasible solution. Consequently, $\delta_{\min} \coloneqq \max\{\delta_{\min}^k\}_{k=1}^K$ is the smallest parameter such that each solution path is well-defined on $[\delta_{\min}, \infty)$.

After solving (P_k), which implies using only the samples $S \setminus S_k$, we evaluate the generalization properties of the k-th solution path by calculating its generalization error w.r.t. S_k. Then, we take the average over all folds to calculate the cross-validation error

$$\varepsilon : [\delta_{\min}, \infty) \to \mathbb{R} \ , \quad \varepsilon(\delta) \coloneqq \frac{1}{K} \sum_{k=1}^K \|A^{I_k} x_k(\delta) - b_{I_k}\|_\infty. \tag{37}$$

The best parameter according to the cross-validation scheme is then

$$\delta_{\mathrm{cv}} \coloneqq \operatorname*{argmin}_{\delta \in [\delta_{\min}, \infty)} \varepsilon(\delta) \ , \quad \text{and} \quad \varepsilon_{\mathrm{cv}} \coloneqq \varepsilon(\delta_{\mathrm{cv}}) \tag{38}$$

is the associated minimal error. Finally, now using the entire set of samples, we solve (36) with $\delta = \delta_{\mathrm{cv}}$ to obtain our estimate $\hat{x}_{\mathrm{cv}}$ of the linear predictor.

The numerical example in Figure 2 is constructed as follows: We perform 10-fold cross-validation on a set of $m = 100$ samples and with $n = 50$. To that end, we generate $A \in \mathbb{R}^{m \times n}$ and a 5-sparse ground truth vector $x^\dagger \in \mathbb{R}^n$ with i.i.d. entries $a_{ij}, x_j^\dagger \sim \mathcal{N}(0, 1)$ randomly (in case of $x^\dagger$, this applies to the non-zero entries) as well as a noise vector $\eta \in \mathbb{R}^m$ with i.i.d. entries $\eta_i \sim \mathcal{U}_{[-\delta^\dagger, \delta^\dagger]}$ with $\delta^\dagger = 1$. Then, we set $b \coloneqq Ax^\dagger + \eta$ and apply the described cross-validation scheme. To validate the goodness of the resulting estimate $\hat{x}_{\mathrm{cv}}$, we compare its ℓ_1-norm distance to the ground truth $d_{\mathrm{cv}} \coloneqq \|\hat{x}_{\mathrm{cv}} - x^\dagger\|_1$ to the distances $d^\dagger \coloneqq \|\hat{x}^\dagger - x^\dagger\|_1$ and $d^* \coloneqq \|\hat{x}^* - x^\dagger\|_1$, where $\hat{x}^\dagger$ is the solution of (36) that results if we use the true parameter $\delta^\dagger$ and $\hat{x}^*$ is the particular point on the solution path of (36) that has the smallest distance to the ground truth (the latter distance is not explicitly marked in the figure, but recognizable as the minimum of the respective function).[3]

[3] Although the entire solution path of (36) with $\delta = \delta_{\min}$ is implicitly availabe through the iterates of ℓ_1-HOUDINI (the breakpoints) and the associated values of the homotopy parameter, we restrict the computation of the minimal distance d^* to the breakpoints themselves. Hence, the "true" d^* is possibly a bit lower than the one we computed. The same statement applies to the computation of δ_{cv} and $\varepsilon_{\mathrm{cv}}$: Although $\varepsilon(\delta)$ can be computed via the respective solution paths, we only evaluate $\varepsilon(\delta)$ where one of the K paths has a breakpoint. As a consequence, the minimum of ε might not be located exactly at the computed δ_{cv}.

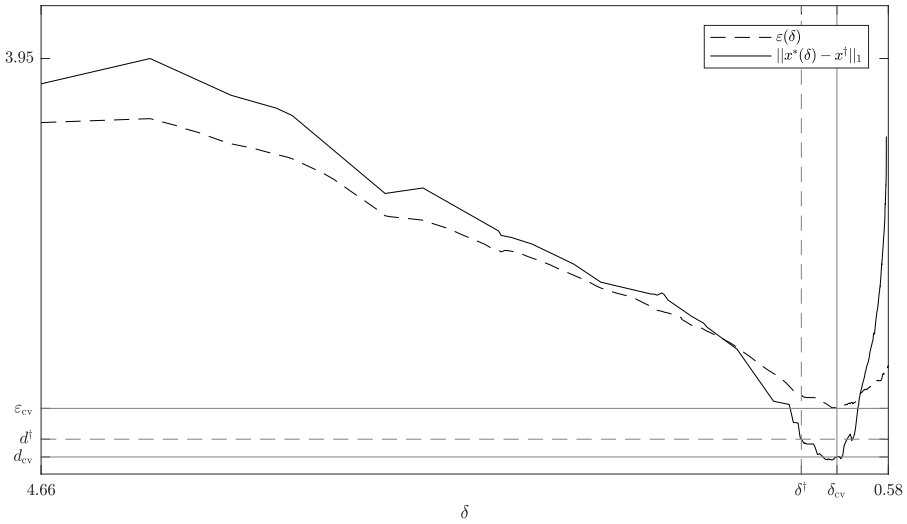

Fig. 2 Exemplary results of the discussed cross-validation scheme. Horizontal axis: linear scale. Vertical axis: log scale. The dashed line represents the cross-validation error $\varepsilon(\delta)$ which attains its minimum $\varepsilon_{\text{cv}} = 1.0076$ at $\delta_{\text{cv}} = 0.8285$. The ℓ_1-distance of the associated linear predictor $\hat{x}_{\text{cv}}$ to the ground truth is $d_{\text{cv}} = 0.5983$. The distance of the predictor $\hat{x}^\dagger$ related to the true parameter $\delta^\dagger = 1$ is $d^\dagger = 0.7480$ and thus comparatively higher. The minimal distance $d^* = 0.8285$ of the solution path to the ground truth is attained at $\hat{x}^*$, which is the solution associated with $\delta^* = 0.8535$.

Example 4.3 (Run-time comparison) We compare our homotopy method for (P_δ) with the state-of-the-art commercial LP solver Gurobi applied to the LP reformulation

$$\min_{x^\pm \in \mathbb{R}^n} \mathbb{1}^\top x^+ + \mathbb{1}^\top x^- \quad \text{s.t.} \quad -\delta \cdot \mathbb{1} \leq Ax^+ - Ax^- - b \leq \delta \cdot \mathbb{1},\ x^+ \geq 0,\ x^- \geq 0$$

(note that this formulation is equivalent to the one stated in Section 1.1.1, which contains slack variables). We experiment with two variants of our ℓ_1-Houdini algorithm: We solve the subproblems either with the specialized active-set methods from Section 3, or with the same LP solver (i.e., Gurobi) with which we solve the above LP reformulation of (P_δ).

Our ℓ_1-Houdini is implemented in Matlab. From the same wrapper code to read instance data, we call either ℓ_1-Houdini to solve for the entire homotopy path, or call Gurobi (via its Matlab interface).

The test instances are constructed from the "L1-Testset" described in [18] (available online via the last author's or the SPEAR project homepage). It contains over 500 instances A, $\bar{x}$ and $b = A\bar{x}$ for the pure Basis Pursuit problem (BP) such that $\bar{x}$ is the known unique optimal solution. Based on the following observation, we can (for a given δ) construct new vectors $\hat{b}$ such that $\bar{x}$ is optimal for the instance of (P_δ) specified by A, $\hat{b}$ and δ: If $\bar{x}$ is optimal for (BP), then there is $\bar{y}$ with $-A^\top \bar{y} \in \operatorname{Sign}(\bar{x})$. If we set $\hat{b} = A\bar{x} - \delta \operatorname{Sign}(\bar{y})$,

then $A\bar{x} \in \hat{b} + \delta\,\mathrm{Sign}(\bar{y})$ and still $-A^\top \bar{y} \in \mathrm{Sign}(\bar{x})$, i.e., $(\bar{x}, \bar{y})$ is a primal dual optimal pair for (P_δ).

We used this observation to construct instances for (P_δ) (the optimal dual certificates $\bar{y}$ for the associated (BP) problem have been computed with the help of [18, Sections 4 and 5 (particularly, Theorem 5.1)]). For the present experiments, we randomly choose two instances for each of the matrix sizes $512 \times \{1024, 1536, 2048, 4096\}$ and $1024 \times \{2048, 3072, 4096, 8192\}$ (cf. [18, Table II])—one in which $\bar{x}$ has nonzero entries of high dynamic range (i.e., the absolute values of the non-zero entries of $\bar{x}$ span several orders of magnitude), and one with low dynamic range, see [18, Sec. 4] for details. This way, we end up with 16 instances, which we will identify by their L1-Testset number (the instance details can be found in the table accompanying the test instance download package). The δ-values were chosen uniformly at random from the interval $[0.1, 5]$ for each instance. Moreover, since we observed that the $\bar{y}$ constructed in the above-mentioned ways are fully dense (which, by complementary slackness, implies that the primal active sets in the respective optimal solutions are also as large as possible), we computed a second set of $\hat{b}$-vectors using other dual certificates that were computed, aiming at sparsity, by solving problems of the form

$$\min_{y \in \mathbb{R}^m} \|y\|_1 \quad \text{s.t.} \quad -A^\top y \in \mathrm{Sign}(\bar{x}).$$

Thus, we have 32 instances in total, with pairs sharing the same instance number, A, δ and optimal solution $\bar{x}$ but having different measurement vectors $\hat{b}$.

The running time results of our experiments (conducted in MATLAB 2016a, using Gurobi 6.5.2, on Ubuntu with an Intel® Core™ i7-4550U CPU @ 1.50GHz × 4 processor) are summarized in Table 1.

In the majority of cases, we observed that ℓ_1-HOUDINI (available on the first author's homepage) using specialized active-set methods for the subproblems is considerably faster than ℓ_1-HOUDINI using Gurobi (32 out of 32 instances) and even faster than Gurobi used as standalone LP solver (20 out of 32 instances). Another comparison suggests that Gurobi used as standalone solver is usually faster than ℓ_1-HOUDINI using Gurobi for the subproblems (30 out of 32 instances). (Nevertheless, note that ℓ_1-HOUDINI generates the entire solution path w.r.t. the homotopy parameter, whereas solving the LP formulation of (P_δ) solely yields a solution for the final parameter δ.)

In particular, it seems beneficial to use ℓ_1-HOUDINI when $|\mathcal{S}|$ is small (i.e., when the optimal solution x^* is relatively sparse). This is a natural feature of our method since the sparsity of the iterates has direct impact on the size of the arising subproblems. Analogously, the size of the primal active set $|\mathcal{A}|$ directly affects the size of the subproblems. Our experiments show that solving the very same instance with smaller optimal active set (induced by a modified measurement vector $\hat{b}$) causes an average speedup of 29.6% and 37.6% using ℓ_1-HOUDINI with active-set methods and Gurobi for the subproblems, respectively. In contrast, using Gurobi as standalone LP solver induces an average speedup of 9.4%.

inst. no.	$m \times n$	δ	$\lvert\mathcal{S}\rvert$	$\lvert\mathcal{A}\rvert$	time ℓ_1-HOUDINI (active set)	time ℓ_1-HOUDINI (Gurobi)	time Gurobi
7	512×1024	4.09	34	512	0.98	2.58	0.46
				72	0.53	2.64	0.46
485	512×1024	4.54	51	512	1.80	103.35	1.26
				96	1.22	–	1.09
25	512×1536	0.72	14	512	0.24	3.60	0.83
				31	0.24	3.77	0.81
319	512×1536	4.58	22	512	0.42	15.92	1.63
				43	0.29	10.40	1.53
228	512×2048	3.20	51	512	5.86	–	1.13
				141	3.79	–	0.98
338	512×2048	0.58	20	512	0.79	–	1.93
				45	0.44	16.13	1.42
74	512×4096	1.47	10	512	0.20	18.36	1.26
				38	0.16	1.06	1.22
347	512×2048	2.78	10	512	0.14	8.32	1.25
				32	0.09	0.86	1.21
239	1024×2048	4.79	84	1024	0.77	2.13	0.07
				148	0.82	2.02	0.07
357	1024×2048	4.83	27	1024	1.91	–	3.51
				55	0.80	38.83	2.77
99	1024×3072	0.87	18	1024	0.91	19.65	3.36
				47	0.76	17.49	3.42
527	1024×3072	4.86	99	1024	26.57	–	1.79
				234	16.46	–	1.59
263	1024×4096	4.79	97	1024	36.53	–	2.99
				245	27.36	437.62	2.69
416	1024×4096	2.48	26	1024	2.47	–	6.85
				60	1.33	50.41	3.99
148	1024×8192	4.02	20	1024	1.41	23.21	5.34
				64	1.34	20.67	5.29
421	1024×8192	0.80	9	1024	0.82	–	5.12
				43	0.42	–	5.27

Table 1 Runtime comparison of ℓ_1-HOUDINI against Gurobi. In case of the instances that are marked with a –, the algorithm stopped prematurely because Gurobi failed to solve one of the subproblems.

In additional experiments with instances from the L1-Testset, we observed that ℓ_1-HOUDINI is also competitive in the Basis Pursuit setting ($\delta = 0$). To that end, we compared our method with ℓ_1-Homotopy (see [2]), one of the fastest methods according to [18], and again with Gurobi as standalone LP solver. The results are subsumed in Table 2. They show that in this special case, ℓ_1-HOUDINI is not as fast as ℓ_1-Homotopy but in most cases still considerably faster than Gurobi.

It is worth mentioning that we also performed testruns on some of the large-scale instances with sparse coefficient matrices from the L1-Testset, where ℓ_1-HOUDINI was competitive as well and often considerably faster than Gurobi (even though Gurobi is tuned for sparse data); details are omitted for brevity.

inst. no.	$m \times n$	$\lvert\mathcal{S}\rvert$	time ℓ_1-HOUDINI	time ℓ_1-Homotopy	time Gurobi
7	512×1024	34	0.83	0.06	0.57
485	512×1024	34	2.09	0.08	1.50
25	512×1536	34	0.28	0.03	0.72
319	512×1536	34	0.47	0.05	1.45
228	512×2048	34	6.37	0.18	0.88
338	512×2048	34	0.90	0.06	2.05
74	512×4096	34	0.23	0.05	1.36
347	512×4096	34	0.16	0.05	1.36
239	1024×2048	34	0.84	0.45	0.08
357	1024×2048	34	2.06	0.11	3.53
99	1024×3072	34	0.94	0.09	3.21
527	1024×3072	34	27.75	0.74	1.83
263	1024×4096	34	35.93	1.08	3.24
416	1024×4096	34	2.53	0.17	9.03
148	1024×8192	34	1.56	0.23	7.23
421	1024×8192	34	0.95	0.15	7.20

Table 2 Runtime comparison of ℓ_1-HOUDINI (active set) against ℓ_1-Homotopy (with regularization parameter $\tau = 10^{-9}$) and Gurobi, all applied to the case $\delta = 0$.

Example 4.4 (Dantzig selector problems) As mentioned earlier, the Dantzig selector problem is a special case of (P_δ), in which the constraint is replaced by $\|A^\top(Ax - b)\|_\infty \leq \delta$. A specific homotopy scheme for this problem that uses the special structure (i.e., a quadratic postive semidefinite operator in the constraint), called *Primal Dual pursuit* (PDP), was proposed in [3]. In Table 3, we compare our method against PDP and Gurobi on Dantzig selector problems constructed according to [9]. The comparison shows that the runtimes of ℓ_1-HOUDINI and PDP often lie in the same magnitude while the respective runtimes of Gurobi are significantly larger. The better performance of PDP may also lie in an efficient implementation, but one reason is that PDP makes efficient use of the special structure in that it always uses linear systems instead of linear programs for the update step (which is not possible in the general non-square case). We can further observe that ℓ_1-HOUDINI is fastest in case $m > n$, which is of interest in many *machine learning* applications, where the number of training examples is much larger than the number of features.

Example 4.5 (Multiple changes in the support can be necessary) As mentioned earlier in Subsection 1.1.2, ℓ_1-HOUDINI allows for multiple changes per iteration in the primal and dual supports, while the PDP homotopy algorithm from [3] is designed for exactly one change per iteration. Simple examples show that indeed multiple changes in the primal and dual support can occur and cannot be handled by successive updates for all new indices. Consider an instance of the Dantzig selector with $m = n = 2$,

$$A = \begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix}, \quad b = \begin{pmatrix} 2 \\ 0 \end{pmatrix} \tag{39}$$

$m \times n$	δ	$\lvert\mathcal{S}\rvert$	$\lvert\mathcal{A}\rvert$	time ℓ_1-HOUDINI	time PDP	time Gurobi
1024×1024	0.39	66	66	0.16	0.13	2.06
1024×1024	0.51	152	152	1.08	0.67	2.15
1024×2048	0.27	69	69	0.34	0.24	8.22
1024×2048	0.39	166	166	2.93	1.52	8.56
2048×1024	0.35	65	66	0.18	0.24	2.11
2048×1024	0.54	128	128	0.45	0.52	2.16
2048×2048	0.29	64	64	0.42	0.37	8.32
2048×2048	0.39	130	130	0.82	0.82	8.45

Table 3 Runtime comparison of ℓ_1-HOUDINI (active set) against PDP and Gurobi, all applied to the Dantzig selector problem (DS_δ).

and an arbitrary $\delta < 2$. Now initialize $x^0 = 0$ and $\delta^0 = \|A^\top Ab\|_\infty = 2$ and note that $A^\top A = 2I_2$ and $A^\top(Ax^0 - b) = -(2,2)^\top$, i.e., both primal constraints are active at the starting point. If we restrict ourselves to a single change per iteration, then the search direction in the first primal update will either be $d^1 \propto (1,0)^\top$ or $d^1 \propto (0,1)^\top$. In the first case, it holds that $A^\top Ad^1 \propto (1,0)^\top$ while we have $A^\top Ad^1 \propto (0,1)^\top$ in the second case. Either way, a step in direction d^1 does only shrink one of the two active constraints while the value of the respective other constraint remains unchanged. As each step $x^0 + t^1 d^1$ with $t^1 > 0$ causes a decrease $\delta^1 = \delta^0 - t^1 < 2$ of the homotopy parameter as well, we conclude that each non-trivial step in direction d^1 induces an infeasible solution. As a consequence, we see that, in order to move away from x^0 towards the next breakpoint, we need to allow multiple changes per iteration in the support. A similar situation can be constructed in any dimension.

Thus, allowing multiple support changes is necessary, although empirically, such events appear to be a rare occurrence in practice. Nevertheless, it turns out that ℓ_1-HOUDINI captures the type of situation described in Example 4.5 by treating primal and dual supports and active sets separately and allowing multiple changes in each iteration (however, it could be possible that the theory for PDP can be adapted to handle these situations, too, but we do not discuss such possibilities any further here).

5 Extensions and Conclusion

Our algorithm can be extended straightforwardly to treat the more general problem class

$$\min_{x \in \mathbb{R}^n} \|x\|_1 \quad \text{s.t.} \quad \alpha \leq Ax - b \leq \beta,\ Dx = d, \tag{40}$$

assuming w.l.o.g. that $\alpha < \beta$ and that the feasible set is nonempty.

To that end, first observe that we can rewrite

$$\alpha \leq Ax - b \leq \beta \quad \Leftrightarrow \quad \underbrace{\alpha - \tfrac{\alpha+\beta}{2}}_{=-\gamma} \leq Ax - \underbrace{\left(b + \tfrac{\alpha+\beta}{2}\right)}_{=:\tilde{b}} \leq \underbrace{\beta - \tfrac{\alpha+\beta}{2}}_{=:\gamma};$$

since $\alpha < \beta$, $\gamma_i \neq 0$ for all i, we can scale each row by $\hat{\delta}/\gamma_i$ for an arbitarily chosen $\hat{\delta} > 0$ and obtain

$$\begin{aligned} & -G\gamma \leq G(Ax - \tilde{b}) \leq G\gamma \\ \Leftrightarrow \quad & -\hat{\delta}\mathbb{1} \leq GAx - G\tilde{b} \leq \hat{\delta}\mathbb{1} \quad \Leftrightarrow \quad \|GAx - G\tilde{b}\|_\infty \leq \hat{\delta}, \end{aligned}$$

where $G = \hat{\delta}\,\mathrm{Diag}(1/\gamma_1, \ldots, 1/\gamma_m)$. Thus, in the absence of equality constraints $Dx = d$, (40) can be recast into the form (P_δ) directly.

However, such an equality constraint is obviously equivalent to requiring $\|Dx - d\|_\infty \leq 0$. Therefore, we can extend the homotopy treatment of problem (P_δ) (where we drive the homotopy parameter down to the target δ-value) to (40) by linking the homotopy parameter δ to the bounds from both ℓ_∞-norm constraints derived from (40) and reducing it all the way to zero. For $\delta = 0$, the homotopy constraints $\|GAx - G\tilde{b}\|_\infty \leq \hat{\delta} + \delta$ and $\|Dx - d\|_\infty \leq \delta$ then correspond exactly to those of (40). Considering two ℓ_∞-norm constraints simultaneously, and the offset $\hat{\delta}$ in one of them, leads to minor simple modifications to the update subproblems in our algorithm; we omit the straightforward details for brevity. Note that for $\delta = \delta^0 := \max\{\|d\|_\infty, \|b\|_\infty - \hat{\delta}\}$, $x = 0$ is an optimal solution for the problem

$$\min_{x \in \mathbb{R}^n} \|x\|_1 \quad \text{s.t.} \quad \|GAx - G\tilde{b}\|_\infty \leq \hat{\delta} + \delta,\ \|Dx - d\|_\infty \leq \delta$$

and thus provides the starting point for our method in the present context.

Further generalizations are likely possibly. For instance, it should be possible to modify the algorithm to treat one-sided bounds ($\alpha_i = -\infty$ or $\beta_i = +\infty$); then, in particular, the case of nonnegative variables could be handled directly, and by means of a standard variable split into the respective positive and negative parts, general linear objective functions (with all coefficients nonzero) could be replaced by the ℓ_1-norm w.r.t. appropriately rescaled variables. Since a thorough investigation of such considerations goes beyond the scope of the present paper, we leave it open for future research.

A Active-Set Method for Linear Programs

A.1 Optimality Conditions for Linear Programs

Let $c \in \mathbb{R}^n$, $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, $D \in \mathbb{R}^{k \times n}$, $e \in \mathbb{R}^k$ and $\sigma \in \{\pm 1\}^n$.[4] We consider the linear program

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & c^\top x \\ \text{s.t.} \quad & Ax = b \\ & Dx \geq e \\ & \mathrm{Diag}(\sigma)x \geq 0 \end{aligned} \tag{41}$$

[4] At this point, we use the standard notation for linear programs. The labels A and b appear as well in the preceding sections. However, they do not have the same meaning here.

and assume that it is feasible and bounded. By the well-known KKT conditions (see, e.g., [21, Theorem 12.1]), x^* is an optimal solution of (41) if and only if there exist Lagrange multipliers $\lambda \in \mathbb{R}^m$, $\mu \in \mathbb{R}^k$ and $\nu \in \mathbb{R}^n$ such that the following conditions hold:

$$Ax^* = b \tag{42a}$$
$$Dx^* \geq e \tag{42b}$$
$$\mathrm{Diag}(\sigma)x^* \geq 0 \tag{42c}$$
$$A^\top \lambda + D^\top \mu + \mathrm{Diag}(\sigma)\nu = c \tag{42d}$$
$$\mu \odot (Dx^* - e) = 0 \tag{42e}$$
$$\nu \odot x^* = 0 \tag{42f}$$
$$\mu \geq 0 \tag{42g}$$
$$\nu \geq 0. \tag{42h}$$

A.2 General Theme

Suppose that $x^\ell \in \mathbb{R}^n$ is feasible for (41), i.e., it satisfies (42a)-(42c). Then, there exist non-empty sets $\mathcal{A} \subseteq \{1, \ldots, k\}$ and $\mathcal{S} \subseteq \{1, \ldots, n\}$ such that

$$D^{\mathcal{A}} x^\ell = e_{\mathcal{A}}, \quad D^{\mathcal{A}^c} x^\ell > e_{\mathcal{A}^c}, \quad x^\ell_{\mathcal{S}^c} = 0 \quad \text{and} \quad |x^\ell_{\mathcal{S}}| > 0.$$

We refer to $\mathcal{A}$ as the *active set* and further to $\mathcal{S}$ as the *support* of x^ℓ. In the context of (42e) and (42f), necessarily $\mu_{\mathcal{A}^c} = 0$ and $\nu_{\mathcal{S}} = 0$ in case x^ℓ is an optimal solution to (41). The following Lemma exploits this fact and provides alternative optimality conditions for (41).

Lemma A.1 *A point x^ℓ is an optimal solution to (41) if and only if it is feasible and there exist $\lambda \in \mathbb{R}^m$ and $\mu_{\mathcal{A}} \in \mathbb{R}^{|\mathcal{A}|}$ such that*

$$A_{\mathcal{S}}^\top \lambda + (D_{\mathcal{S}}^{\mathcal{A}})^\top \mu_{\mathcal{A}} = c_{\mathcal{S}}, \tag{43a}$$
$$\mathrm{Diag}(\sigma_{\mathcal{S}^c})(c_{\mathcal{S}^c} - A_{\mathcal{S}^c}^\top \lambda - (D_{\mathcal{S}^c}^{\mathcal{A}})^\top \mu_{\mathcal{A}}) \geq 0 \quad \textit{and} \tag{43b}$$
$$\mu_{\mathcal{A}} \geq 0. \tag{43c}$$

Proof It can easily be shown that the conditions in Lemma A.1 are equivalent to conditions (42a)–(42h) with $\mu_{\mathcal{A}^c} = 0$, $\nu_{\mathcal{S}} = 0$ and

$$\nu_{\mathcal{S}^c} = \mathrm{Diag}(\sigma_{\mathcal{S}^c})(c_{\mathcal{S}^c} - A_{\mathcal{S}^c}^\top \lambda - (D_{\mathcal{S}^c}^{\mathcal{A}})^\top \mu_{\mathcal{A}}). \tag{44}$$

□

Starting from x^ℓ, our goal is to approach a solution of (41) by generating *descent directions* ξ that preserve the active set as well as the support and, should this not be possible, by changing these sets appropriately. We repeat these steps until we finally identify $\mathcal{A}$, $\mathcal{S}$, λ and $\mu_{\mathcal{A}}$ satisfying (43a)–(43c).

A.3 Descent Directions and Blocking Constraints

If there exists a solution of the linear system

$$\begin{bmatrix} A_{\mathcal{S}} \\ D_{\mathcal{S}}^{\mathcal{A}} \\ c_{\mathcal{S}}^\top \end{bmatrix} \xi_{\mathcal{S}} = \begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix} \quad \text{and} \quad \xi_{\mathcal{S}^c} = 0, \tag{45}$$

then it holds for arbitrary $\alpha > 0$ that

$$A(x^\ell + \alpha\xi) = b, \quad D^{\mathcal{A}}(x^\ell + \alpha\xi) = e_{\mathcal{A}} \quad \text{and} \quad x^\ell_{\mathcal{S}^c} + \alpha\xi_{\mathcal{S}^c} = 0. \tag{46}$$

The largest $\alpha > 0$ such that also

$$D^{\mathcal{A}^c}(x^\ell + \alpha\xi) \geq e_{\mathcal{A}^c} \quad \text{and} \quad \mathrm{Diag}(\sigma_{\mathcal{S}})(x^\ell_{\mathcal{S}} + \alpha\xi_{\mathcal{S}}) \geq 0 \tag{47}$$

is given by

$$\alpha = \min \left(\min_{\substack{i \in \mathcal{A}^c \\ d_i^\top \xi < 0}} \frac{e_i - d_i^\top x^\ell}{d_i^\top \xi}, \min_{\substack{j \in \mathcal{S} \\ \sigma_j \xi_j < 0}} -\frac{x_j}{\xi_j} \right). \tag{48}$$

Note that $0 < \alpha < \infty$ since we assumed that (41) is bounded. The sets

$$\mathcal{A}^+ = \left\{ i \in \mathcal{A}^c : d_i^\top (x^\ell + \alpha\xi) = e_i \right\} \quad \text{and} \quad \mathcal{S}^- = \left\{ j \in \mathcal{S} : x^\ell_j + \alpha\xi_j = 0 \right\} \tag{49}$$

are the index sets where the minimum is attained, i.e., the sets of *blocking constraints*. Each $i \in \mathcal{A}^+$ joins the active set and each $j \in \mathcal{S}^-$ leaves the support if we perform the step $\alpha\xi$. Consequently, we update $x^{\ell+1} = x^\ell + \alpha\xi$, $\mathcal{A} = \mathcal{A} \cup \mathcal{A}^+$ and $\mathcal{S} = \mathcal{S} \setminus \mathcal{S}^-$.

A.4 Lagrange Multipliers

If there is no direction according to (45), then zero is an optimal solution of

$$\begin{aligned} \min_{\xi_{\mathcal{S}} \in \mathbb{R}^{|\mathcal{S}|}} \quad & c_{\mathcal{S}}^\top \xi_{\mathcal{S}} \\ \text{s.t.} \quad & \begin{bmatrix} A_{\mathcal{S}} \\ D^{\mathcal{A}}_{\mathcal{S}} \end{bmatrix} \xi_{\mathcal{S}} = 0 \end{aligned} \tag{50}$$

Employing KKT conditions again, we see that there exist λ and $\mu_{\mathcal{A}}$ satisfying (43a). For the case that λ and $\mu_{\mathcal{A}}$ additionally satisfy (43b)–(43c), Lemma A.1 states that x^ℓ is an optimal solution.

Otherwise, with $\nu_{\mathcal{S}^c}$ according to (44), there exists at least one index $i \in \mathcal{A}$ such that $\mu_i < 0$ or $j \in \mathcal{S}^c$ such that $\nu_j < 0$. We select the smaller of both values and set $\mathcal{A} = \mathcal{A} \setminus \{i\}$ or $\mathcal{S} = \mathcal{S} \cup \{j\}$, respectively. Then, we search a new direction according to Subsection A.3.

A.5 Feasibility of Generated Directions

In the context of the previous section, suppose that $\mu_i < 0$ and we set $\mathcal{A} = \mathcal{A} \setminus \{i\}$. Afterwards, we go back to (45) and find a direction ξ. It holds that

$$\begin{aligned} -1 \overset{(45)}{=} c_{\mathcal{S}}^\top \xi_{\mathcal{S}} &\overset{(43a)}{=} (A_{\mathcal{S}}^\top \lambda + (D^{\mathcal{A}}_{\mathcal{S}})^\top \mu_{\mathcal{A}} + (D^i_{\mathcal{S}})^\top \mu_i)^\top \xi_{\mathcal{S}} \\ &= \lambda^\top A_{\mathcal{S}} \xi_{\mathcal{S}} + \mu_{\mathcal{A}}^\top D^{\mathcal{A}}_{\mathcal{S}} \xi_{\mathcal{S}} + \mu_i D^i_{\mathcal{S}} \xi_{\mathcal{S}} \\ &\overset{(45)}{=} \mu_i d_i^\top \xi. \end{aligned} \tag{51}$$

It follows that $d_i^\top \xi = -\mu_i^{-1} > 0$. Consequently, it holds that $d_i^\top (x^\ell + \alpha\xi) > e_i$ and the step $\alpha\xi$ preserves the property of $\mathcal{A}$ exactly reflecting the set of active constraints. An analogous statement holds if we update $\mathcal{S} = \mathcal{S} \cup \{j\}$ prior to finding a direction ξ. In that case, we obtain $\sigma_j \xi_j = -\nu_j^{-1} > 0$.

Note that, if we found $\mu_{\{i,i'\}} < 0$ for distinct indices $i, i' \in \mathcal{A}$ and set $\mathcal{A} = \mathcal{A} \setminus \{i, i'\}$, we would not necessarily get $d_i^\top \xi > 0$ and $d_{i'}^\top \xi > 0$. Repeating the above reasoning only

shows $(\mu_i d_i + \mu_{i'} d_{i'})^\top \xi > 0$. The same holds if we have $\nu_{\{j,j'\}} < 0$ or $\mu_i < 0$ and $\nu_j < 0$. Therefore, we do not change more than one index before we search for a new direction. However, it can occur that we do not immediately find a new direction after changing one index in $\mathcal{A}$ or $\mathcal{S}$. In that case, we have to determine Lagrange multipliers repeatedly and change $\mathcal{A}$ and $\mathcal{S}$ until we are able to find a new direction. This situation needs to be handled with care in order to correctly keep track of $\mathcal{A}$ and $\mathcal{S}$. We capture this aspect in Appendix A.7.

A.6 Fixing New Support Variables

Equation (51) further shows that, if we replace $c_\mathcal{S}^\top \xi_\mathcal{S} = -1$ by $d_i^\top \xi = 1$ in (45), this implies $c^\top \xi = \mu_i < 0$. The resulting system is

$$\begin{bmatrix} A^\mathcal{S} \\ D_\mathcal{A}^\mathcal{S} \\ D_i^\mathcal{S} \end{bmatrix} \xi_\mathcal{S} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}. \tag{52}$$

Numerically, there is no obvious gain in the replacement of one equation. Essentially, the new constraint specifies $d_i^\top \xi = 1$. The same reasoning for the case that $j \in \mathcal{S}$ was recently added to the support shows that by dropping $c_\mathcal{S}^\top \xi_\mathcal{S} = -1$ and fixing $\xi_j = \sigma_j$, we obtain $c^\top \xi = \nu_j < 0$. Considering the numerical effort, this can be beneficial since we not only drop a constraint but also reduce the number of variables in the system. The result is

$$\begin{bmatrix} A^{\mathcal{S}\setminus\{j\}} \\ D_\mathcal{A}^{\mathcal{S}\setminus\{j\}} \end{bmatrix} \xi_{\mathcal{S}\setminus\{j\}} = -\sigma_j \begin{pmatrix} A^j \\ D_\mathcal{A}^j \end{pmatrix}. \tag{53}$$

A.7 Algorithm and Implementation of ℓ_1-HOUDINI

Algorithm 2 illustrates the iterative scheme discussed in Appendix A.2–A.6. Additionally, we assume that an initial direction ξ is provided as input since this is the situation we are faced with in Section 3.

The conditional statement beginning in Step 11 considers two special cases. In that context, $\mathcal{A}^-$ is the set of indices that were consecutively removed from the active set in Steps 25–32 and $\mathcal{S}^+$ is the set of indices that were consecutively added to the support. It can occur that $|\mathcal{A}^-| + |\mathcal{S}^+| > 1$ in case we do not find a direction in Step 5 in a positive number of consecutive iterations.

The first case is $\alpha = 0$ which can occur if $|\mathcal{A}^-| + |\mathcal{S}^+| > 1$ and there exists $i \in \mathcal{A}^-$ such that $d_i^\top \xi < 0$ or $\sigma_j \xi_j < 0$ for some $j \in \mathcal{S}^+$. The respective indices are re-added to $\mathcal{A}$ and re-removed from $\mathcal{S}$, respectively, before trying to find a new feasible direction.

In the second case, if $\alpha > 0$ and $|\mathcal{A}^-| + |\mathcal{S}^+| > 1$, we can still have $i \in \mathcal{A}^-$ with $d_i^\top \xi = 0$ or $\sigma_j \xi_j = 0$ for some $j \in \mathcal{S}^+$. Consequently, the i-th constraint stays active and j does not join the support after a step in direction ξ. We adapt $\mathcal{A}$ and $\mathcal{S}$ accordingly. Since we have performed a non-zero step, we moreover reset $\mathcal{A}^-$ and $\mathcal{S}^+$.

Table 5 puts the primal update from Section 3.1 into the context of Algorithm 2. Notice that problem (15) needs to be reformulated as a minimization problem in order to have the form (41). Table 4 does the same for the dual update from Section 3.2.

In both the primal and the dual case we applied some easy sign substitutions in order to bring (43a) into a simple form. Of course, the respective inverse substitutions appear in the formulas for $\mu_\mathcal{S}$ and $\nu_{\mathcal{S}^c}$, respectively.

Moreover, we used that during the primal update $\operatorname{sign}(y_{I_D}^{k+1}) = \operatorname{sign}(A^{I_D} \xi^\ell - b_{I_D})$ throughout.

Algorithm 2: Active-Set Method for LPs.

Input: $c \in \mathbb{R}^n$, $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, $D \in \mathbb{R}^{k \times n}$, $e \in \mathbb{R}^k$, $\sigma \in \{\pm 1\}^n$, feasible $x^0 \in \mathbb{R}^n$ and associated sets $\mathcal{A}$ and $\mathcal{S}$, initial direction ξ

Output: solution x^* to problem (41)

$\ell \leftarrow 0$
$\mathcal{A}^- \leftarrow \emptyset$
$\mathcal{S}^+ \leftarrow \emptyset$
while *not stopped* **do**
  if *a solution* ξ *of* (45) *exists* **then**
    $\alpha \leftarrow$ step size according to (48)
    $x^{\ell+1} \leftarrow x^\ell + \alpha\xi$
    $(\mathcal{A}^+, \mathcal{S}^-) \leftarrow$ blocking constraints according to (49)
    $\mathcal{A} \leftarrow \mathcal{A} \cup \mathcal{A}^+$
    $\mathcal{S} \leftarrow \mathcal{S} \setminus \mathcal{S}^-$
    if $\alpha = 0$ **then**
      $\mathcal{A}^- \leftarrow \mathcal{A}^- \setminus \mathcal{A}^+$
      $\mathcal{S}^+ \leftarrow \mathcal{S}^+ \setminus \mathcal{S}^-$
    else
      if $|\mathcal{A}^-| + |\mathcal{S}^+| > 1$ **then**
        $\mathcal{A} \leftarrow \mathcal{A} \cup \{i \in \mathcal{A}^- : d_i^\top \xi = 0\}$
        $\mathcal{S} \leftarrow \mathcal{S} \setminus \{j \in \mathcal{S}^+ : \xi_j = 0\}$
      $\mathcal{A}^- \leftarrow \emptyset$
      $\mathcal{S}^+ \leftarrow \emptyset$
    $\ell \leftarrow \ell + 1$
  else
    $(\mu_{\mathcal{A}}, \nu_{\mathcal{S}^c}) \leftarrow$ Lagrange multipliers according to (43a) and (44)
    $i^- \leftarrow \operatorname{argmin}_{i \in \mathcal{A}} \mu_i$
    $j^+ \leftarrow \operatorname{argmin}_{j \in \mathcal{S}^c} \nu_j$
    if $\mu_{i^-} \geq 0$ *and* $\nu_{j^+} \geq 0$ **then**
      return $x^* = x^\ell$
    else if $\mu_{i^-} < \nu_{j^+}$ **then**
      $\mathcal{A} \leftarrow \mathcal{A} \setminus \{i^-\}$
      $\mathcal{A}^- \leftarrow \mathcal{A}^- \cup \{i^-\}$
    else
      $\mathcal{S} \leftarrow \mathcal{S} \cup \{j^+\}$
      $\mathcal{S}^+ \leftarrow \mathcal{S}^+ \cup \{j^+\}$

References

1. Appa, G., Smith, C.: On L_1 and Chebyshev estimation. Math. Programming **5**, 73–87 (1973). URL https://doi.org/10.1007/BF01580112
2. Asif, M.S.: Primal Dual Pursuit—A Homotopy Based Algorithm for the Dantzig Selector. Master's thesis, Georgia Institute of Technology (2008)

$\mathcal{S}$	I_D
$\mathcal{A}$	$J_D \setminus J_P$
$A_{\mathcal{S}}$	$(-A_{J_P}^{I_D})^\top$
$D_{\mathcal{S}}^{\mathcal{A}}$	$(A_{J_D \setminus J_P}^\top \psi^\ell) \odot (-A_{J_D \setminus J_P}^{I_D})^\top$
$c_{\mathcal{S}}$	$-\operatorname{sign}(A^{I_D} x^k - b_{I_D})$
(45)	$(A_{J_D}^{I_D})^\top e_{I_D} = 0$
	$\operatorname{sign}(A^{I_D} x^k - b_{I_D})^\top e_{I_D} = 1$
(43a)	$A_{J_D}^{I_D} \hat{d}_{J_D} = -\operatorname{sign}(A^{I_D} x^k - b_{I_D})$
$\mu_{\mathcal{A}}$	$-(A_{J_D \setminus J_P}^\top \psi^\ell) \odot \hat{d}_{J_D \setminus J_P}$
$\sigma_{\mathcal{S}^c}$	$\operatorname{sign}(A^{I_P \setminus I_D} x^k - b_{I_P \setminus I_D})$
$c_{\mathcal{S}^c}$	$-\operatorname{sign}(A^{I_P \setminus I_D} x^k - b_{I_P \setminus I_D})$
$A_{\mathcal{S}^c}$	$(-A_{J_P}^{I_P \setminus I_D})^\top$
$D_{\mathcal{S}^c}^{\mathcal{A}}$	$(A_{J_D \setminus J_P}^\top \psi^\ell) \odot (-A_{J_D \setminus J_P}^{I_P \setminus I_D})^\top$
$\nu_{\mathcal{S}^c}$	$-\operatorname{sign}(A^{I_P \setminus I_D} x^k - b_{I_P \setminus I_D}) \odot A^{I_P \setminus I_D} \hat{d} - \mathbb{1}$

Table 4 Active-Set Implementation of the Dual Update.

$\mathcal{S}$	$J_P \cup \{t\}$
$\mathcal{A}$	$I_P \setminus I_D$
$A_{\mathcal{S}}$	$\begin{bmatrix} A_{J_P}^{I_D} & \operatorname{sign}(y_{I_D}^{k+1}) \end{bmatrix}$
$D_{\mathcal{S}}^{\mathcal{A}}$	$\begin{bmatrix} -\operatorname{sign}(A^{I_P \setminus I_D} \xi^\ell - b_{I_P \setminus I_D}) \odot A_{J_P}^{I_P \setminus I_D} & -\mathbb{1} \end{bmatrix}$
$c_{\mathcal{S}}$	$(0, -1)^\top$
(45)	$A_{J_P}^{I_P} d_{J_P} = -\operatorname{sign}(A^{I_P} \xi^\ell - b_{I_P})$
(43a)	$(A_{J_P}^{I_P})^\top \hat{e}_{I_P} = 0$
	$\operatorname{sign}(A^{I_P} \xi^\ell - b_{I_P})^\top \hat{e}_{I_P} = 1$
$\mu_{\mathcal{A}}$	$\operatorname{sign}(A^{I_P \setminus I_D} \xi^\ell - b_{I_P \setminus I_D}) \odot \hat{e}_{I_P \setminus I_D}$
$\sigma_{\mathcal{S}^c}$	$A_{J_P \setminus J_D}^\top y^{k+1}$
$c_{\mathcal{S}^c}$	0
$A_{\mathcal{S}^c}$	$A_{J_D \setminus J_P}^{I_D}$
$D_{\mathcal{S}^c}^{\mathcal{A}}$	$-\operatorname{sign}(A^{I_P \setminus I_D} \xi^\ell - b_{I_P \setminus I_D}) \odot A_{J_D \setminus J_P}^{I_P \setminus I_D}$
$\nu_{\mathcal{S}^c}$	$-(A_{J_D \setminus J_P}^\top y^{k+1}) \odot A_{J_D \setminus J_P}^\top \hat{e}$

Table 5 Active-Set Implementation of the Primal Update.

3. Asif, M.S., Romberg, J.: Dantzig selector homotopy with dynamic measurements. In: Proc. SPIE 7246, Computational Imaging VII, 72460E (2009)
4. Asif, M.S., Romberg, J.: On the LASSO and Dantzig selector equivalence. In: Proc. CISS. IEEE (2010)
5. Blum, L.: A New Simple Homotopy Algorithm for Linear Programming I. Journal of Complexity **4**, 124–136 (1988)
6. Brauer, C., Gerkmann, T., Lorenz, D.A.: Sparse Reconstruction of Quantized Speech Signals. In: Proc. ICASSP. IEEE (2016)
7. Cai, T., Liu, W.: A Direct Estimation Approach to Sparse Linear Discriminant Analysis. Journal of the American Statistical Association **106**(496), 1566–1577 (2011)
8. Cai, T., Liu, W., Luo, X.: A Constrained ℓ_1 Minimization Approach to Sparse Precision Matrix Estimation. Journal of the American Statistical Association **106**(494), 594–607 (2011)
9. Candés, E.J., Tao, T.: The Dantzig Selector: Statistical Estimation When p Is Much Larger than n. The Annals of Statistics **35**(6), 2313–2351 (2007)
10. Chen, S.S., Donoho, D.L., Saunders, M.A.: Atomic Decomposition by Basis Pursuit. SIAM Journal on Scientific Computing **20**(1), 33–61 (1998)
11. Dantzig, G.B.: Linear Programming and Extenstions. Princeton University Press (1963)
12. Donoho, D.L.: Compressed Sensing. IEEE Transactions on Information Theory **52**(4), 1289–1306 (2006)
13. E. J. Candés, T.T.: Decoding by linear programming. IEEE Transactions on Information Theory **51**(12), 4203–4215 (2005)
14. Eldar, Y.C., Kutyniok, G. (eds.): Compressed Sensing. Theory and Applications. Cambridge University Press (2012)
15. Jacques, L., Hammond, D.K., Fadili, J.M.: Dequantizing Compressed Sensing: When Oversampling and Non-Gaussian Constraints Combine. IEEE Transactions on Information Theory **57**(1), 559–571 (2011)
16. James, G.M., Radchenko, P., Lv, J.: DASSO: connections between the Dantzig selector and lasso. Journal of the Royal Statistical Society: Series B (Statistical Methodology) **71**(1), 127–142 (2009)
17. Kohavi, R.: A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. In: International Joint Conference on Artificial Intelligence (IJCAI) (1995)
18. Lorenz, D.A., Pfetsch, M.E., Tillmann, A.M.: Solving Basis Pursuit: Heuristic Optimality Check and Solver Comparison. ACM Transactions on Mathematical Software **41**(2), Art. No. 8 (2015)
19. Mairal, J., Yu, B.: Complexity Analysis of the Lasso Regularization Path (2012). arXiv preprint arXiv:1205.0079
20. Nazareth, J.L.: The Homotopy Principle and Algorithms for Linear Programming. SIAM Journal on Optimization **1**(3), 316–332 (1991)
21. Nocedal, J., Wright, S.J.: Numerical Optimization, 2nd edn. Springer (2006)
22. Osborne, M.R., Presnell, B., Turlach, B.A.: A new approach to variable selection in least squares problems. IMA Journal of Numerical Analysis **20**, 389–404 (2000)
23. Pang, H., Zhao, T., Vanderbei, R.J., Liu, H.: A Parametric Simplex Approach to Statistical Learning Problems. Unpublished manuscript (2015). URL http://www.princeton.edu/ rvdb/tex/PSM/PSM.pdf
24. Rockafellar, R.T.: Convex Analysis. Princeton University Press (1970)
25. S. Foucart, H.R.: A Mathematical Introduction to Compressive Sensing. Birkhäuser (2013)
26. Schrijver, A.: Theory of Linear and Integer Programming. John Wiley & Sons (1986)
27. Späth, H.: Mathematical algorithms for linear regression. Computer Science and Scientific Computing. Academic Press, Inc., Boston, MA (1992). Translated and revised from the 1987 German original by the author
28. Stiefel, E.: Über diskrete und lineare tschebyscheff-approximationen. Numerische Mathematik **1**(1), 1–28 (1959). DOI 10.1007/BF01386369. URL https://doi.org/10.1007/BF01386369
29. Vanderbei, R.J.: Linear Programming: Foundations and Extensions, 2nd edn. Kluwer Academic Publishers (2001)

30. Zheng, S., Liu, W.: An experimental comparison of gene selection by Lasso and Dantzig selector for cancer classification. Computers in Biology and Medicine **41**(11), 1033–1040 (2011)