\textbf{\(\ell_1\)-HOUDINI: A New Homotopy Method for \(\ell_1\)-Minimization}

Christoph Brauer*  Dirk Lorenz*  Andreas Tillmann†

* TU Braunschweig, Germany  † RWTH Aachen University, Germany

\section*{Problem and Optimality Conditions}

- Given \(A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m\) and \(\delta \geq 0\), we consider the problem
  \[
  \min_{x \in \mathbb{R}^n} \|x\|_1 \quad \text{s.t.} \quad \|Ax - b\|_\infty \leq \delta. \tag{P_\delta}
  \]

- It is well-known that \(x^*\) is an optimal solution of (P_\delta) if and only if there exists a \(y^*\) such that
  \[
  -A^T y^* \in \partial \|x^*\|_1 \quad \text{and} \quad Ax^* - b \in \partial \|y^*\|_1. \tag{1}
  \]

- Each such \(y^*\) is by construction an optimal solution to the dual problem of (P_\delta), which is
  \[
  \max_{y \in \mathbb{R}^n} -b^T y - \delta \|y\|_1 \quad \text{s.t.} \quad \|A^T y\|_\infty \leq 1. \tag{D_\delta}
  \]

\section*{Basic Idea}

- We solve a sequence of problems (P_\delta)_k=0,\ldots,K with
  \[
  \|b\|_\infty = \delta_0 > \delta_1 > \cdots > \delta_K = \delta.
  \]

- The starting point \((x^0, y^0) = (0, 0)\) is an optimal pair for (P_\delta).

- The transition from an optimal pair \((x^k, y^k)\) for (P_\delta) to an optimal pair \((x^{k+1}, y^{k+1})\) for (P_\delta+1) can be done in two steps:
  \[
  \text{Dual Update } U_D : \text{Fix } x^k \text{ and } \delta^k \text{ in (2) and search an appropriate } y^{k+1} \neq y^k \text{ such that the conditions stay valid at } (x^k, y^{k+1}) \text{ and } \delta^{k+1}. \]

- \[
  \left\{ \begin{array}{l}
  \text{Dual Update } U_D : \text{Fix } y^{k+1} \text{ in (2) and search } x^{k+1} \neq x^k \text{ and } \delta^{k+1} < \delta^k \text{ such that the conditions stay satisfied at } (x^{k+1}, y^{k+1}) \text{ and } \delta^{k+1} \end{array} \right.
  \]

\section*{Properties}

- After \(K \leq (3m+n+1)/2\) consecutive dual and primal updates, the method terminates yielding an optimal pair \((x_\delta, y_\delta)\) for (P_\delta).

- The solution path of (P_\delta) is continuous piecewise linear. Our method implicitly generates an optimal solution for each problem (P_\delta) with \(\delta \leq \delta \leq \|b\|_\infty\).

- The linear programs in \(U_D\) and \(U_P\) can be tackled by an arbitrary LP solver. We propose an active set approach that covers two essential aspects:
  1. The iterates \(y^k\) and \(x^k\) are feasible starting points for \(U_D\) and \(U_P\), respectively.
  2. Lagrange multipliers certifying optimality of \(y^{k+1}\) in \(U_D\) qualify as an initial search direction at \(x^k\) in \(U_P\), and vice versa.

\section*{Partitioned Optimality Conditions}

- For a thorough understanding of the conditions (1), we define
  \[
  S := \{ j : x_j^* \neq 0 \}, \quad \Omega := \{ i : |A_i^T x^* - b_i| = \delta \}, \quad \Sigma := \{ j : |A_j^T y^*| = 1 \}, \quad \Omega := \{ i : y_i^* \neq 0 \},
  \]
  \[
  (\text{primal support}) \quad (\text{primal active set}) \quad (\text{dual active set}) \quad (\text{primal support})
  \]

- The optimality conditions (1) are then equivalent to
  \[
  \left\{ \begin{array}{l}
  -A_\delta^T y^* = \text{sign}(x^*_\Omega) \\
  x^*_\Omega - b_\Omega = \delta \text{ sign}(y^*_\Sigma) \\
  -\|x^*_\Omega - b_\Omega\|_1 \leq -\|A_\delta^T y^*\|_\infty \\
  x^*_{\Omega^c} = 0
  \end{array} \right. \tag{2}
  \]

\section*{Dual Update \(U_D\)}

- \(S\) and \(\Omega\) now denote the support and active set of \(x^\delta\).

- We solve the following linear program with \(|\Omega|\) bounded variables and \(2n - |\Sigma|\) constraints to obtain a new dual solution:
  \[
  \begin{array}{ll}
  y^{k+1}_W \in \arg \min_{y_W \in \mathbb{R}^{|\Omega|}} & -\text{sign}(A_W^T x^\delta - b_W) \cdot y_W \\
  \text{s.t.} & (A_W^T x^\delta - b_W) \cdot y_W \leq \|y_W\|_1 \\
  & (\delta^k - t) \cdot y_W \leq \|x^\delta - b_W\|_1 \\
  & \|y_W\|_1 = 0
  \end{array}
  \]

\section*{Primal Update \(U_P\)}

- In the following, \(\Omega\) and \(\Sigma\) denote the support and active set of \(y^{k+1}\).

- For the primal update, we solve the following linear program with \(|\Sigma| + 1\) bounded variables and \(2m - |\Omega|\) constraints:
  \[
  (x^{k+1}_\Sigma, y^{k+1}) \in \arg \max_{(x_{\Sigma}, t) \in \mathbb{R}^{|\Sigma|} \times \mathbb{R}} t \\
  \text{s.t.} & A_\Omega^T x^\delta - b_\Omega = \delta^k - t \text{ sign}(y^{\delta+1}_\Omega) \\
  & (\delta^k - t) I \leq A_\Sigma^T x^\delta - b_\Sigma \leq (\delta^k - t) I \\
  & A_\Sigma^T y^{\delta+1} \cdot x_\Sigma \leq 0 \\
  & t \leq \delta^k - \delta
  \]

- \(x^{k+1}_\Sigma := 0\)

- \(\delta^{\delta+1} := \delta^k - \delta^{k+1}\)

- The choice of the objective functions in \(U_D\) and \(U_P\) is motivated by a theorem of the alternative and plays a key role in view of finite termination.
Exemplary Solution Path

![Exemplary Solution Path](image)

The first part of the comparison shows that the runtimes of \( \ell_1 \)-Houini and PDP \([4]\) are significantly larger than the number of features. Applied to the empirical data from \([5]\), we can further observe that the algorithm needed 9 iterations to solve the problem. Horizontal labels display the value of the homotopy parameter \( \delta \) after each iteration. The plots represent the solution paths of \( x_j^r \) for \( j = 1, \ldots, 12 \). The optimal solution has 6 nonzero entries.

### Runtime and Accuracy Comparison for the Dantzig Selector [4]

<table>
<thead>
<tr>
<th>inst.</th>
<th>runtime in seconds</th>
<th>( | x^r |_1 )</th>
<th>constraint violation</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \ell_1 )-Houini</td>
<td>PDP</td>
<td>Gurobi</td>
<td>( \ell_1 )-Houini</td>
</tr>
<tr>
<td>1</td>
<td>0.19</td>
<td>0.14</td>
<td>2.22</td>
</tr>
<tr>
<td>2</td>
<td>1.02</td>
<td>0.64</td>
<td>2.36</td>
</tr>
<tr>
<td>3</td>
<td>0.34</td>
<td>0.27</td>
<td>8.93</td>
</tr>
<tr>
<td>4</td>
<td>2.74</td>
<td>1.48</td>
<td>9.19</td>
</tr>
<tr>
<td>5</td>
<td>0.21</td>
<td>0.26</td>
<td>2.26</td>
</tr>
<tr>
<td>6</td>
<td>0.47</td>
<td>0.52</td>
<td>2.35</td>
</tr>
<tr>
<td>7</td>
<td>0.44</td>
<td>0.41</td>
<td>9.11</td>
</tr>
<tr>
<td>8</td>
<td>0.84</td>
<td>0.86</td>
<td>9.22</td>
</tr>
<tr>
<td>9</td>
<td>0.03</td>
<td>0.02</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td>10</td>
<td>0.03</td>
<td>0.02</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td>11</td>
<td>0.02</td>
<td>0.1</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td>12</td>
<td>0.04</td>
<td>0.02</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td>13</td>
<td>0.02</td>
<td>—</td>
<td>0.03</td>
</tr>
<tr>
<td>14</td>
<td>0.21</td>
<td>3.47</td>
<td>0.52</td>
</tr>
<tr>
<td>15</td>
<td>176.76</td>
<td>5.52</td>
<td>1.11</td>
</tr>
</tbody>
</table>

The table shows that \( \ell_1 \)-Houini is fastest, with \( \ell_1 \)-Houini \([4]\) being the only algorithm that works with high accuracy on the whole test set. The number of training examples is much larger than the number of features. Applied to the empirical data from \([5]\), \( \ell_1 \)-Houini is the fastest algorithm in the majority of cases, while PDP fails to find an optimal solution in three out of seven cases. The table shows that \( \ell_1 \)-Houini is the only algorithm that works with high accuracy on the whole test set.

### References


