Efficiently Optimizing over (Non-Convex) Cones via Approximate Projections

Michael B. Cohen
Massachusetts Institute of Technology
micohen@mit.edu

Chinmay Hegde
Iowa State University
chinmay@iastate.edu

Stefanie Jegelka
Massachusetts Institute of Technology
stefje@mit.edu

Ludwig Schmidt
Massachusetts Institute of Technology
ludwigs@mit.edu

Abstract

Constrained least squares is a ubiquitous optimization problem in machine learning, statistics, and signal processing. While projected gradient descent is usually an effective algorithm for solving constrained least squares at scale, the projection operator is often the computational bottleneck, especially for complicated constraints. To circumvent this limitation, we extend recent work on approximate projections to a significantly broader range of constrained least squares problems. Our new variant of projected gradient descent is able to utilize approximate projections for any condition number and any conic constraint set (including non-convex cones).

1 Introduction

Constrained optimization is a core element in machine learning, statistics, and signal processing. Over the past decade, these fields have developed a sophisticated understanding of various constraint sets, both from an optimization perspective and from a statistical point of view. The prototypical problem in this area is constrained least squares: given a data or measurement matrix $X \in \mathbb{R}^{n \times d}$, observations $y \in \mathbb{R}^n$, and a constraint set $C \subseteq \mathbb{R}^d$, the goal is to find a minimizer of

$$\min_{\theta \in C} \|X \theta - y\|_2^2. \tag{1}$$

For various constraint sets $C$, this optimization problem can represent important estimators such as (kernel) ridge regression, the Lasso, and nuclear norm minimization. The success of these methods leads to a fundamental question:

For what constraint sets can we solve least squares efficiently?

We make progress on this question by introducing a new version of projected gradient descent that utilizes approximate projections [8]. Approximate projections relax the usual notion of projecting onto a set by allowing a relative approximation error. Prior work has shown that approximate projections can be significantly faster than their exact counterparts for many natural constraint sets (see Table 1). In some settings such as graph sparsity, exactly projecting onto the constraint set is an NP-hard problem, while an approximate projection can still be computed in nearly-linear time [9].

Although the aforementioned papers establish significantly faster running times, their results are limited to the RIP (Restricted Isometry Property) setting that is common in the compressive sensing literature [7]. From an optimization point of view, this is a highly restrictive assumption as it requires the condition number of the objective function, i.e., the quadratic loss $\|X \theta - y\|_2^2$, to be close to 1. Moreover, prior results require the condition number to improve even more as the projections become more approximate [8, 10].

In this paper, we generalize the approximate projection framework along multiple directions:

- We generalize the definitions of approximate projections to any conic constraint set (even non-convex cones).

<table>
<thead>
<tr>
<th>Constraint set</th>
<th>Best known time complexity of an exact projection</th>
<th>Best known time complexity of an approximate projection</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sparsity</td>
<td>$O(d)$</td>
<td>$O(d)$</td>
</tr>
<tr>
<td>Low-rank matrices</td>
<td>$O(d^{1.5})$</td>
<td>$\tilde{O}(r \cdot d)$ [10]</td>
</tr>
</tbody>
</table>

Table 1: The time complexity of exact and approximate projections for various constraint sets. The variable $d$ denotes the size of the input (the dimension of the vector $\theta$). For low-rank matrices, the stated time complexities are for matrices with dimension $\sqrt{d} \times \sqrt{d}$. To simplify the expressions, we omit logarithmic factors in the running times. In all cases beyond simple sparsity, approximate projections are significantly faster (assuming $P \neq NP$, even by a super-polynomial amount).

- We introduce a new variant of projected gradient descent (2PHASE-PGD) that can be combined with approximate projections for any condition number and any conic constraint set.
- We prove that 2PHASE-PGD achieves optimal statistical guarantees (up to constant factors) in all relevant problem parameters, including the condition number.

Interestingly, our new algorithm 2PHASE-PGD combines ideas from both the Frank-Wolfe algorithm and projected gradient descent. The outer loop of 2PHASE-PGD is essentially projected gradient descent, and each iteration of the outer loop calls an inner subroutine that is very similar to the Frank-Wolfe algorithm. The combination of these two parts allows our algorithm to utilize approximate projections without restrictions on the constraint sets and condition numbers.

### 1.1 Related work

Due to the vast literature on constrained least squares estimators, we only mention the most closely related works here.

The combination of approximate projections that we utilize in our work was first introduced in the model-based compressive sensing setting [8], where the goal is to add additional sparsity-based constraints to compressive sensing algorithms [4]. The authors later generalized their approach to general union-of-subspace models that include low-rank matrices [10]. However, all of their algorithms rely on very well conditioned objective functions (the RIP setting). Our algorithm lifts this restriction and applies to any condition number (and any conic constraint set).

The paper [14] also considers projected gradient descent for general constraint sets that can be non-convex. However, the goals of the paper are somewhat different from ours: the authors focus on establishing sharp constants for isotropic measurement setups or data matrices, which corresponds to the well-conditioned regime (RIP). In contrast, our focus is on the regime where the condition number can be arbitrary. Moreover, our algorithm works with approximate projections.

Finally, the paper [13] also considers iterative thresholding methods (i.e., projected gradient descent) in the regime where the condition number can be arbitrary. However, the sample complexity does not match the optimal rates achieved by the Lasso estimator because the statistical rate has a quadratic dependence on the condition number. Our two-phase variant of projected gradient descent addresses this shortcoming. Moreover, our algorithm works with approximate projections and arbitrary conic constraint sets.

### 2 Results

We state our results in the common setting where we assume restricted strong convexity and smoothness [1]. In contrast to the classical (global) notions of strong convexity and smoothness, the restricted counterparts only hold over the constraint set (or a suitably relaxed version of the constraint set). This weakened assumption is crucial because the concentration phenomena in statistical settings are only sufficient to guarantee strong convexity and smoothness over a subset of the entire parameter space.
Definition 1 (Restricted smoothness). Let $C$ be a cone. Then a differentiable convex function $f$ has restricted smoothness $L$ over $C$ if, for every point $x \in \mathbb{R}^d$ and every vector $u \in C$, 
\[
\langle \nabla f(x + u) - \nabla f(x), u \rangle \leq L\|u\|_2^2.
\]
Note that for a quadratic function $\frac{1}{2}\|Ax - b\|_2^2$, this is equivalent to $\|Au\|_2^2 \leq L\|u\|_2^2$ for every $u \in C$.

Definition 2 (Restricted strong convexity). Let $C$ be a cone. Then a differentiable convex function $f$ has restricted strong convexity $\ell$ over $C$ if, for every point $x \in \mathbb{R}^d$ and every vector $u \in C$, 
\[
f(x + u) \geq f(x) + \langle \nabla f(x), u \rangle + \frac{\ell}{2}\|u\|_2^2.
\]
For a quadratic function $\frac{1}{2}\|Ax - b\|_2^2$, this is equivalent to $\|Au\|_2^2 \geq \ell\|u\|_2^2$ for every $u \in C$.

2.1 Approximate projections

Next, we introduce our new definitions of “head” and “tail” approximations, which are two complementary notions of approximate projections. These relaxed projections are the only way our algorithm 2PHASE-PGD interfaces with the constraint sets.

Definition 3 (Head approximation). Let $C^*, C_H \subseteq \mathbb{R}^d$ be two cones and let $c_H \in \mathbb{R}$. Then an $(C^*, C_H, c_H)$-head approximation satisfies the following property. Given any vector $g \in \mathbb{R}^d$, the head approximation returns a unit vector $\theta \in C_H \cap \mathbb{R}^{d-1}$ such that 
\[
\langle g, \theta \rangle \geq c_H \cdot \max_{\theta' \in C^* \cap \mathbb{R}^{d-1}} \langle g, \theta' \rangle.
\]

Definition 4 (Tail approximation). Let $C^*, C_T \subseteq \mathbb{R}^d$ be two cones and let $c_T \in \mathbb{R}$. Then an $(C^*, C_T, c_T)$-tail approximation satisfies the following property. Given any vector $\theta^m \in \mathbb{R}^d$, the tail approximation returns a vector $\theta \in C_T$ such that 
\[
\|\theta^m - \theta\|_2 \leq c_T \cdot \min_{\theta' \in C_T} \|\theta^m - \theta'\|_2.
\]

In addition to a relaxed approximation guarantee ($c_H < 1$ and $c_T > 1$, respectively), the above definitions also allow for approximation in the output sets $C_H$ and $C_T$. As long as $C_H$ and $C_T$ are comparable to $C^*$ (say sparsity $2 \cdot s$ instead of sparsity $s$), this relaxation affects the sample complexity only by constant factors, yet enables polynomially faster algorithms in cases such as graph sparsity [9].

2.2 Main results

Before we state our main result, we briefly introduce some notation. For two sets $C_1$ and $C_2$, we denote the Minkowski sum by $C_1 + C_2$. For an integer $m$, we write $m \times C$ for the $m$-wise Minkowski sum of a set with itself (i.e., $C + C + C + \cdots + C$ a total of $m$ times).

Theorem 5 (2PHASE-PGD). There is an algorithm 2PHASE-PGD with the following properties. Assume that 2PHASE-PGD is given

- a $(C^*, C_T, c_T)$-tail approximation such that $\theta^* \in C^*$, and
- a $(C^* - C_T, C_H, c_H)$-head approximation.

Then let 
\[
k = \Theta \left( \frac{(1 + c_T)^2 \cdot L_H}{c_H^2 \cdot \ell_{all}} \right)
\]
and assume that $f$ has restricted smoothness $L_H$ over $C_H$ and restricted strong convexity $\ell_{all}$ over the sum of $C^*$ with the negations of $C_T$ and $k$ copies of $C$, i.e., $\ell_{all} = \ell_{C^* - C_T - k \times C}$.

For a given $\varepsilon > 0$ and $R > \|\theta^*\|_2$, 2PHASE-PGD then returns an estimate $\hat{\theta}$ such that 
\[
\|\hat{\theta} - \theta^*\|_2 \leq \max \left( 64 \cdot \frac{(1 + c_T)^2}{\ell_{all}} \max_{\theta' \in (C^* - C_T - 1/c_H \times C) \cap \mathbb{R}^{d-1}} \langle \nabla f(\theta^*), \theta' \rangle, \varepsilon R \right).
\]

The time complexity of 2PHASE-PGD is dominated by $O(\log 1/\varepsilon)$ calls to the tail approximation and $O(k \log 1/\varepsilon)$ calls to the head approximation and the gradient oracle, respectively.

\footnote{To avoid confusion: in this paper, $C^*$ does not denote a polar cone, but instead the constraint set corresponding to the solution $\theta^*$.}
Since the theorem above is somewhat technical, we briefly instantiate it in the standard compressive sensing setting to illustrate the bounds. Due to space constraints, we unfortunately defer a more thorough discussion of the various parameters to the full version of the paper.

Let \( C^*, C_H, C_T \) each be the set of \( s \)-sparse vectors. Since we can project onto \( s \)-sparse vectors in linear time, we have \( c_H = c_T = 1 \). Hence our algorithm requires restricted smoothness and strong convexity only over sets that are \( O(s) \)-sparse, for which we can invoke standard results for RIP matrices [7]. The RIP setting also implies that \( l_1 \approx l_H \approx 1 \). Together, this yields the standard bound \( \|\hat{\theta} - \theta^*\|_2 \leq O(\|e\|_2) \) for a noise vector \( e \) after a sufficient number of iterations (so that the \( \epsilon R \) terms is negligible).

### 3 Algorithm

Algorithms 1 and 2 contain the two main parts of our new algorithm. As mentioned before, Algorithm 1 closely resembles the Frank Wolfe algorithm (conditional gradient method). Algorithm 2 is essentially projected gradient descent, but with INNERPHASE instead of a standard gradient step. We defer a more detailed discussion of these algorithms to the full version of the paper.

#### Algorithm 1 INNERPHASE

1. **function** INNERPHASE\((\theta^{in}, r, \rho)\)
2. \( k \leftarrow \left\lceil \frac{8L_H}{\rho c_H^2} \right\rceil \)
3. \( \theta^{(0)} \leftarrow 0 \)
4. **for** \( i \leftarrow 0, \ldots, k - 1 \) **do**
5. \( g^{(i+1)} \leftarrow \text{HEADAPPROX}(\nabla f(\theta^{in} + \theta^{(i)})) \)
6. \( \tilde{\theta}^{(i+1)} \leftarrow \frac{r}{c_H} \cdot \frac{g^{(i+1)}}{\|g^{(i+1)}\|_2} \)
7. \( \theta^{(i+1)} \leftarrow \gamma^{(i)} \cdot \tilde{\theta}^{(i+1)} + (1 - \gamma^{(i)}) \cdot \theta^{(i)} \)
8. **return** \( \theta^{in} + \theta^{(k)} \)

#### Algorithm 2 2PHASE-PGD

1. **function** 2PHASE-PGD\((R, \varepsilon)\)
2. \( T \leftarrow \left\lceil \log_2 \frac{1}{\varepsilon} \right\rceil \)
3. \( \theta^{(0)} \leftarrow \tilde{0} \)
4. \( r^{(0)} \leftarrow R \)
5. **for** \( t \leftarrow 0, \ldots, T - 1 \) **do**
6. \( \tilde{\theta}^{(t+1)} \leftarrow \text{INNERPHASE}(\theta^{(t)}, r^{(t)}) \)
7. \( \theta^{(t+1)} \leftarrow \text{TAILAPPROX}(\tilde{\theta}^{(t+1)}) \)
8. **if** \( \|\theta^{(t+1)} - \theta^{(t)}\|_2 > \frac{3}{2} r^{(t)} \) **then return** \( \hat{\theta} \leftarrow \theta^{(t)} \)
9. \( r^{(t+1)} \leftarrow \frac{r^{(t)}}{2} \)
10. **return** \( \hat{\theta} \leftarrow \theta^{(T)} \)
References


