Relax-and-split method for nonsmooth nonconvex problems.

Peng Zheng
Department of Applied Mathematics
University of Washington
Seattle, WA 98195-3925, USA

Aleksandr Aravkin
Department of Applied Mathematics
University of Washington
Seattle, WA 98195-3925, USA

Editor:

Abstract

We develop and analyze a new ‘relax-and-split’ (RS) approach for compositions of separable nonconvex nonsmooth functions with linear maps. RS uses a relaxation technique together with partial minimization, and brings classic techniques including direct factorization, matrix decompositions, and fast iterative methods to bear on nonsmooth nonconvex problems. We also extend the approach to trimmed nonconvex-composite formulations; the resulting Trimmed RS (TRS) can fit models while detecting outliers in the data.

We then test RS and TRS on a diverse set of applications: (1) phase retrieval, (2) stochastic shortest path problems, (3) semi-supervised classification, and (4) new clustering approaches. RS/TRS can be applied to models with very weak functional assumptions, are easy to implement, competitive with existing methods, and enable a new level of modeling formulations to be put forward to address emerging challenges in the mathematical sciences.

1. Introduction

Extracting information from large-scale datasets is essential for modern scientific computing and data-driven discovery. Classic techniques such as least squares and direct decompositions (such as the singular value decomposition) demand a prohibitively high degree of data quality, regularity, and homogeneity. Inference in many settings requires robustness to error, enforcement of solution structure, and control of model complexity. These features can be effectively captured using nonsmooth and nonconvex optimization formulations.

In this paper, we consider nonconvex-composite problems:

\[
\min_x f(x) := h(Ax) + g(x),
\]

where \( x \in \mathbb{R}^n \) are decision variables, \( A = [a_1, \ldots, a_m]^T \in \mathbb{R}^{m \times n}, h : \mathbb{R}^m \to \mathbb{R} \) is nonsmooth, nonconvex, and separable, so \( h(Ax) = \sum_{i=1}^m h_i(\langle a_i, x \rangle) \); while \( g : \mathbb{R}^n \to \mathbb{R} \) is convex. We also consider trimmed extensions to robustify such models:

\[
\min_{x,v} \sum_{i=1}^m v_i h_i(\langle a_i, x \rangle) + g(x), \quad \text{s.t. } v \in \Delta_{\tau},
\]

where \( \Delta_{\tau} := \{v : v \in [0, 1]^m, \sum_{i=1}^m v_i = \tau\} \) is the so called capped simplex. The auxiliary variables \( v \) detect \( m - \tau \) outliers amongst the \( m \) observations as the optimization proceeds.
1.1 Examples

We present motivating examples for (1) before reviewing the literature and explaining the contributions. Each example is explained fully in Section 6. Examples 1-3 are not weakly convex, that is, they cannot be convexified by adding a quadratic. Weak convexity is a key property for the convergence theory of competitive methods covered in Section 1.3; RS does not require weak convexity. All of these examples can be robustified against outliers using trimming (2); trimming formulations are discussed at the end of Section 1.3 and the TRS approach for (2) is developed in Section 4.

Example 1 (Sharp phase retrieval) Given a complex matrix $A \in \mathbb{C}^{m \times n}$, the phase retrieval problem attempts to recover the full complex signal $x$ using only moduli $b$:

$$\min_{x \in \mathbb{C}^{n}} \sum_{i=1}^{m} |\langle a_i, x \rangle| - b_i.$$

Example 2 (Semi-Supervised Classification) Logistic regression is a common approach for binary classification; training requires labeled examples. We solve an extended approach that makes use of both labeled and unlabeled data:

$$\min \frac{\lambda}{2} \|x\|^2 + \sum_{i=1}^{l} \log(1 + \exp(-b_i \langle a_i, x \rangle)) + \tau \sum_{i=l+1}^{m} \log(1 + \exp(-|\langle a_i, x \rangle|)),$$

where $a_i$ are features, $b_i$ labels for the first $s$ examples, and remaining $(m - l)$ examples are not labeled. The idea is to separate unlabeled examples as clearly as possible, regardless of which class they fall into.

Example 3 (Stochastic Shortest Path) Given a weighted graph on $n$ nodes, we look for a policy that minimizes expected cost of path to target by selecting between one of two actions at each node. Let $U^k \in \mathbb{R}^{n \times n}$ and $v^k \in \mathbb{R}^n$ be the connectivity graphs and average node costs for $k = 1, 2$. Using the Bellman equation, the problem is formulated as

$$\min_{x \in \mathbb{R}^n} \sum_{i=1}^{n} |\min \{\langle u^1_i, x \rangle + v^1_i - x_i, \langle u^2_i, x \rangle + v^2_i - x_i\}|,$$

where $u^k_i$ is the $i$-th row vector of $U^k$ and $x_i$ is the best expected cost starting from node $i$.

Example 4 (Convex and Nonconvex Clustering) While K-means is the most widely used clustering method, an alternative is to solve the problem

$$\min_{X} \frac{\lambda}{2} \sum_{i=1}^{m} \|x_i - u_i\|^2 + \lambda \sum_{i=1}^{m-1} \sum_{j=i+1}^{m} R([D^i]_{ij})$$

$$s.t. \ [D^i]_{ij} = x_i - x_j$$

where $u_i$ is the reference data points and $X = [x_1, \ldots, x_m]$ are the decision variables, with $R$ a regularization functional that acts to ‘fuse’ columns $X$ into cluster representatives, and $\lambda$ a regularization parameter that effectively controls the number of clusters. Classic approaches use a convex $R$, but we find a nonconvex $R$ has significant advantages.
Table 1: Mapping motivating applications into class (1)

| Example       | $h(z)$         | Linear map | $g(x)$       |
|---------------|----------------|------------|--------------|
| Phase retrieval | $|z| - b$ | $A$ | 0            |
| SS-LR         | $\log(1 + \exp(-|z|))$ | $A$ | $\frac{\lambda}{2} \|x\|^2$ |
| Stoch. path   | $\min\{z - a, z - b\}$ | $U^1, U^2$ | 0            |
| Clustering    | $R(z)$         | $D$ | $\frac{1}{2} \sum_{i=1}^m \|x_i - u_i\|^2$ |

Table 1 maps Examples 1-4 to the templated objective (1). While the only theoretical requirement for $g(x)$ is convexity, in practice we assign simple smooth terms to $g$, so that we can implement fast subproblem solves. We can always take $g(x) = 0$ if necessary, rewriting a problem with multiple terms into a simple composition $h(Ax)$:

$$f_1(Bx) + f_2(x) = h \left( \begin{bmatrix} B \\ I \end{bmatrix} x \right), \quad \text{with} \quad A = \begin{bmatrix} B \\ I \end{bmatrix}, \quad \text{and} \quad h(z_1, z_2) = f_1(z_1) + f_2(z_2).$$

The choice $g(x) = 0$ is allowed by the theory and common in practice.

1.2 RS for Nonconvex Composite Models

The core innovation of this work is to relax (1) and (2) by introducing an auxiliary variable $w$, and then use partial minimization over the original variables to develop efficient algorithms. In particular, we take the following ‘relaxed’ version of (1):

$$\min_{w,x} f_\nu(x, w) := h(w) + \frac{1}{2\nu} \|Ax - w\|^2 + g(x),$$

where $w$ approximates $Ax$, decoupling the linear map from the nonsmooth, nonconvex $h$. The structure of (7) allows a partial minimization scheme. Define

$$g_\nu(w) := \min_x \frac{1}{2\nu} \|Ax - w\|^2 + g(x).$$

Problem (7) is now equivalent to

$$\min_w p_\nu(w) := h(w) + g_\nu(w).$$

Several observations can be made.

- Since $g$ is convex, (8) can be solved efficiently, especially when $g$ is also smooth.
- Conditioning of (9) is independent of $A$ (see Table 2).
- The prox operator of $h$ is easy to apply whenever $h$ is separable.

These points affect the theoretical convergence and practical implementation of RS, and are made precisely in the analysis detailed in Section 3.
Contributions  Our contributions are as follows.

- We develop relaxed models for (1) and (2), which are simple to optimize and very effective across a diverse set of applications (measured by application-specific metrics).

- We derive provably convergent algorithms for these relaxations, obtaining rates under different conditions on $g$ and $h$. In contrast to recent work for nonsmooth nonconvex optimization, we do not assume that $h$ is weakly convex. The new methods thus apply to a broader range of problems than prior art, and can handle e.g. exact phase retrieval and semi-supervised learning.

- We apply the approach to get promising application-specific results:
  - Exact phase retrieval, along with a trimmed robust extension;
  - Semi-supervised classification;
  - New direct approach for the stochastic shortest path problem;
  - A new scalable approach for convex and nonconvex clustering.

1.3 Related Work

Well-known approaches for nonsmooth, nonconvex problems include nonsmooth BFGS (Lewis and Overton, 2009), Gradient Sampling (Burke et al., 2005), and derivative free methods (DFO), see e.g. (Conn et al., 2009). These methods can be applied to problems more general than those in class (1) and (2); but they assume nothing about problem structure, and so there is little chance of scaling them to the semi-supervised SVMs and phase retrieval problems in our numerical examples, which have millions of variables. The lack of structure also limits the available convergence analysis: theoretical grounding for nonsmooth BFGS appears elusive; GS finds Clarke stationary points with unknown speed, while rates for DFO are known and must scale linearly with dimension.

More closely related to this paper is convex-composite optimization, which captures problems in classic nonlinear programming and more recently in large-scale machine learning. The convex-composite class, see e.g. Burke (1985a); Burke and Ferris (1995)) generalizes both smooth and convex functions and is given by

$$\min_{x \in \mathbb{R}^n} \sum_{i=1}^{m} h_i(c_i(x)) + g(x),$$

where $g$ is a closed convex function, $h_i$ are convex and Lipschitz, and $c_i$ are smooth maps. The functions $g$ and $h_i$ provide an inference structure, while the maps $c_i$ encode the data generating mechanism. Examples include exact penalty formulations of nonlinear programs (Nocedal and Wright, 2006, Section 17.2), robust phase retrieval (squared variant) (Duchi and Ruan, 2017b), and matrix factorization (Gillis, 2017). Convex-composite problems have been extensively studied over the years (Cartis et al., 2011; Powell, 1983; Burke, 1985b; Yuan, 1985; Wright, 1990; Fletcher, 1982; Powell, 1984), and have seen significant recent interest (Lewis and Wright, 2015; Drusvyatskiy and Lewis, 2016; Cartis et al., 2011; Nesterov, 2007; Drusvyatskiy and Paquette, 2016; Duchi and Ruan, 2017a).
The problem classes (1) and (2) fall outside of the convex-composite class any time $h$ is both nonsmooth and nonconvex \(^1\). On the other hand, the nonconvex-composite class assumes that the data generating mechanism $Ax$ is linear. An analysis of the natural super-class that allows nonsmooth nonconvex $h$ and nonlinear maps $c$ is left to future work.

Smoothing techniques are closely related to our approach; Moreau-Yosida smoothing (see Section 2) and related method of Nesterov (2005) are at the core of many well-known algorithms, including those of Becker et al. (2011), Yang and Zhang (2011), and Xu et al. (2015). If we partially minimize (7) with respect to $w$ rather than with respect to $x$, we arrive at the problem

$$\min_x h_\nu(Ax) + g(x),$$

with $h_\nu$ analogous to the smoother discussed in Nesterov (2005). However, since $h$ is nonconvex, the function $h_\nu$ may also be nonsmooth and nonconvex (see the right panel of Figure 1), and (11) may be just as difficult to solve as the original problem. Minimizing over $x$ instead leads to analyzable algorithms in the nonconvex-composite setting.

![Figure 1: Moreau-Yosida smoothing for convex and nonconvex functions. The left figure plots smoothers for the convex function $h(x) = |x|$, while the right figure plots smoothers for the function $h(x) = ||x| - 1|$, which is not even weakly convex.](image)

Another line of recent work combines stochastic gradient techniques with nonsmooth optimization (Aravkin and Davis, 2016; Davis and Drusvyatskiy, 2018). These approaches typically require stronger assumptions, such as smoothness or weak convexity of $h$. A function $h$ is $\rho$-weakly convex when $h(\cdot) + \frac{\rho}{2}||\cdot||^2$ is convex. No function with ‘inward kinks’ can be weakly convex, which eliminates every one of our motivating examples.

Finally, we discuss the prior literature on trimmed estimation. Trimmed M-estimators were initially introduced by Rousseeuw (1985) in the context of least-squares regression. Recent work developed statistical theory (Alfons et al., 2013; Yang and Lozano, 2015; Yang et al., 2016) for robust high-dimensional applications, including lasso, graphical lasso, and sparse logistic regression. The Proximal Alternating Linearized Minimization (PALM) method of Bolte et al. (2014) can be used to find trimmed estimators (2) so long as the $h$ functions are smooth and have Lipschitz continuous gradients. Better rates under the

---

1. When $h$ is smooth, $h(Ax)$ is smooth also and hence trivially convex-composite.
same assumptions are achieved by the algorithm of Aravkin and Davis (2016), who study the general formulation

$$\min_{x,v} \sum_{i=1}^{m} v_i h_i(x) + g(x), \quad \text{s.t. } v \in \triangle_{\tau},$$

(12)

where $\tau < m$ is the estimated number of inliers, and the model $h : \mathbb{R}^m \to \mathbb{R}$ is smooth, while $g(x)$ is prox-bounded. Variables $v$ separate inliers from outliers by finding elements $h_i(x)$ that disagree with the consensus, even as the consensus evolves due to updates of $x$. The set $\triangle_{\tau}$, called the capped simplex, as the intersection of the $\tau$-simplex with the unit box, see (2). We extend the RS method to the nonconvex-composite class (2), so that we can trim nonsmooth nonconvex terms. This extension, called trimmed RS (TRS), allows for outlier detection and removal for any of the motivating examples, and we illustrate the power of the approach on the phase retrieval application in Section 6.1.

1.4 Road map

The paper proceeds as follows. RS is developed and analyzed in Section 3. The trimming extension and TRS are presented in Section 4. Practical considerations, including implementation, approximation and refinement, and discussed in Section 5, along with a comparison to the frequently used Alternating Directions Method of Multipliers (ADMM) problem in the convex setting. Detailed descriptions and results for the motivating applications are presented in Section 6. Proofs and technical details are collected in Appendix A.

2. Notation and Preliminaries

In this section, we recall some basic notation that we will use throughout the manuscript. We will follow closely the monographs of Mordukhovich (2006) Rockafellar and Wets (1998).

Euclidean Space. Throughout, we consider a Euclidean space, denoted by $\mathbb{R}^n$, with an inner product $\langle \cdot, \cdot \rangle$ and the induced norm $\| \cdot \|$. Given a linear map $A : \mathbb{R}^n \to \mathbb{R}^m$, the adjoint $A^\top : \mathbb{R}^m \to \mathbb{R}^n$ is the unique linear map satisfying

$$\langle Ax, y \rangle = \langle x, A^\top y \rangle \quad \text{for all } x \in \mathbb{R}^n, y \in \mathbb{R}^m.$$

The operator norm of $A$, defined as $\|A\| := \max_{\|u\| \leq 1} \|Au\|$, coincides with the maximal singular value of $A$ and satisfies $\|A\| = \|A^\top\|$.

Functions and Geometry. The extended-real-line is the set $\overline{\mathbb{R}} := \mathbb{R} \cup \{\pm \infty\}$. The domain and the epigraph of any function $f : \mathbb{R}^d \to \overline{\mathbb{R}}$ are the sets

$$\text{dom } f := \{x \in \mathbb{R}^d : f(x) < +\infty\}, \quad \text{epi } f := \{(x, r) \in \mathbb{R}^d \times \mathbb{R} : f(x) \leq r\}.$$

We say that $f$ is closed if its epigraph, epi $f$, is a closed set. We assume that all functions that we encounter are proper, meaning they have nonempty domains and never take on the value $-\infty$. All the functions we consider in this paper are closed and proper.
**Lipschitz Continuity.** For any map $F : \mathbb{R}^n \rightarrow \mathbb{R}^m$, we set,

$$\text{lip}(F) := \sup_{x \neq y} \frac{\|F(y) - F(x)\|}{\|y - x\|}.$$  

In particular, we say that $F$ is $L$-Lipschitz continuous, for some $L \geq 0$, if the inequality $\text{lip}(F) \leq L$ holds.

**Fréchet and Limiting Subdifferentials.** Consider an arbitrary function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ and a point $\bar{x}$ with $f(\bar{x})$ finite. The Fréchet subdifferential of $f$ at $\bar{x}$, denoted $\hat{\partial}f(\bar{x})$, is the set of all vectors $v$ satisfying

$$f(x) \geq f(\bar{x}) + \langle v, x - \bar{x} \rangle + o(\|x - \bar{x}\|) \quad \text{as} \quad x \rightarrow \bar{x}.$$  

Thus the inclusion $v \in \hat{\partial}f(\bar{x})$ holds precisely when the affine function $x \mapsto f(\bar{x}) + \langle v, x - \bar{x} \rangle$ underestimates $f$ up to first-order near $\bar{x}$.

In general, the limit of Fréchet subgradients $v_i \in \hat{\partial}f(x_i)$, along a sequence $x_i \rightarrow \bar{x}$, may not be a Fréchet subgradient at the limiting point $\bar{x}$. We define the limiting subdifferential of $f$ at $\bar{x}$, denoted $\partial f(\bar{x})$, to comprise all vectors $v$ for which there exist sequences $x_i$ and $v_i$, with $v_i \in \partial f(x_i)$ and $(x_i, f(x_i), v_i) \rightarrow (\bar{x}, f(\bar{x}), v)$.

**Moreau Envelope and Proximal Mapping.** For any function $f$ and real $\nu > 0$, the Moreau envelope and the proximal mapping are defined by

$$f_\nu(x) := \inf_z \left\{ f(z) + \frac{1}{2\nu} \|z - x\|^2 \right\},$$  

(13)

$$\text{prox}_\nu f(x) := \arg\min_z \left\{ f(z) + \frac{1}{2\nu} \|z - x\|^2 \right\}.$$  

(14)

### 3. Convergence Analysis for RS

In this section, we develop and analyze a simple algorithm to find stationary points of the relaxed objective (9).

#### 3.1 Proximal Gradient Method for the Relaxed Objective

Proximal gradient descent method (PGD) is a simple and powerful algorithm in the nonsmooth setting. It requires the objective to be a sum of smooth and ‘prox-friendly’ terms. Problem (9) is naturally viewed this way, since

- $g_\nu$ is smooth and its gradient is Lipschitz continuous, and
- $h$ is prox-friendly; in particular it is separable.

**Theorem 1** Let $g : \mathbb{R}^n \rightarrow \mathbb{R}$ be a proper closed convex function that is bounded below, and $A : \mathbb{R}^n \rightarrow \mathbb{R}^m$ be a linear map. Define function $g_\nu : \mathbb{R}^m \rightarrow \mathbb{R}$ and solution set $x_\nu$ to be,

$$g_\nu(w) = \min_x g(x) + \frac{1}{2\nu} \|Ax - w\|^2,$$

$$x_\nu(w) = \arg\min_x g(x) + \frac{1}{2\nu} \|Ax - w\|^2.$$
For all $x_1, x_2 \in x_\nu(w)$, $x_1 - x_2 \in \text{Null}(A)$. Moreover, $g_\nu$ is convex and $C^1$-smooth, with
\[
\nabla g_\nu(w) = \frac{1}{\nu}(w - Ax), \quad \forall x \in x_\nu(w) \quad \text{and} \quad \text{lip}(\nabla g_\nu) \leq \frac{1}{\nu}.
\]

**Proof** The proof is given in Appendix A. \hfill ■

Theorem 1 establishes the smoothness of $g_\nu$. By the separability of $h$, $\text{prox}_{\gamma h}$ decouples into a set of scalar optimization problems
\[
\text{prox}_{\gamma h}(v) = \arg\min_w \frac{1}{2\gamma} \|w - v\|^2 + h(w) = \left[ \begin{array}{c}
\arg\min_{w_1} \frac{1}{2\gamma} (w_1 - v_1)^2 + h_1(w_1) \\
\vdots \\
\arg\min_{w_m} \frac{1}{2\gamma} (w_m - v_m)^2 + h_m(w_m)
\end{array} \right].
\]

Even though $h$ is nonconvex and nonsmooth, scalar problems are typically easy to solve. To implement the motivating examples, we found closed form solutions for the $\text{prox}$ operators in examples 1, 3, 4, and implemented a Newton method for semi-supervised logistic regression in example 2. Some $h$ require root-finding or bi-section techniques, but due to the separability assumption, these methods need only be applied to scalar problems.

The PGD algorithm is detailed in Algorithm 1.

**Algorithm 1** Proximal Gradient Descent for $h(w) + g_\nu(w)$

**Input:** $w^0$

1. Initialize: $k = 0$
2. while not converge do
3. \hspace{0.5cm} $w^{k+1} \leftarrow \text{prox}_{\nu h}(w^k - \nu \nabla g_\nu(w^k))$
4. \hspace{0.5cm} $k \leftarrow k + 1$
5. end while

**Output:** $w^k$

We can write the $w$-update in Algorithm 1 explicitly:
\[
\text{prox}_{\nu h}(w^k - \nu \nabla g_\nu(w^k)) = \text{prox}_{\nu h}(Ax^k), \quad x^k(w^k) \in \arg\min_x g(x) + \frac{1}{2\nu} \|Ax - w^k\|^2. \quad (15)
\]

In the next section, we analyze the behavior of Algorithm 1 under different assumptions.

### 3.2 Convergence Analysis

The goal for Algorithm 1 is to find the stationary point for (9), defined as follows.

**Definition 2 (Stationary Point)** A point $\bar{w} \in \mathbb{R}^m$ is called a stationary point for (9) if
\[
0 \in \nabla g_\nu(w) + \partial h(w).
\]

Equivalently, we can write
\[
0 \in \left\{ \partial h(\bar{w}) + \frac{1}{\nu} \left( I - A \left( \partial g + \frac{1}{\nu} A^\top A \right)^{-1} A^\top \right) \bar{w} \right\} := S(\bar{w}).
\]
where \((\partial g + \frac{1}{\nu}A^\top A)^{-1}\bar{w}\) is a nonlinear (possibly multi-valued) operator that gives the set of solutions \(x(\bar{w})\) to the problem in (15).

Motivated by this definition, we define the following quantity to measure optimality.

**Definition 3 (Optimality Condition)** We denote

\[
T_\nu(w) = \min \left\{ \|v\|^2 : v \in S(\bar{w}) \right\},
\]

as the optimality condition of (9).

Convergence rates of Algorithm 1 depends on additional assumptions on \(h\) and \(g\), and are summarized in Table 2. All proofs for this section are collected in Appendix A.

| Assumption | Rate of Convergence |
|------------|---------------------|
| Assumption 1 | \(T_\nu^k \leq \frac{2}{\nu k}[p_\nu(w^0) - p_\nu^*]\) |
| Assumption 2 | \(p_\nu(w^k) - p_\nu^* \leq \frac{\|w^k - w^*\|^2}{2\nu(k+1)}\) |
| Assumption 3 | \(\|w^{k+1} - w^*\|^2 \leq \frac{1}{1+\alpha\nu}\|w^k - w^*\|^2\) |
| Assumption 4 | \(\|w^{k+1} - w^*\| \leq \frac{1}{\alpha\nu}\|w^k - w^*\|^2\) |

Table 2: Summary of convergence rates for Algorithm 1. We denote \(\bar{T}_\nu^k\) as the average of quantity (16) in \(k\) steps, namely \(\frac{1}{k} \sum_{i=1}^{k} T_\nu(w_i, w_i^*)\).

3.2.1 General Case

**Assumption 1** \(h\) is prox-bounded, so that there exists a \(\nu\) with \(\text{prox}_{\nu h}(x)\) nonempty for all \(x\) and \(\nu > \nu\); \(g\) is convex.

**Theorem 4** If Assumption 1 holds, the iterates generated by Algorithm 1 satisfy,

\[
\frac{1}{\nu}A(x^{k-1} - x^k) \in \partial h(w^k) + \frac{1}{\nu}(w^k - Ax^k), \quad \text{where} \quad 0 \in \partial g(x^k) + \frac{1}{\nu}A^\top(Ax^k - w^k).
\]

moreover,

\[
\bar{T}_\nu^k := \frac{1}{k} \sum_{i=1}^{k} T_\nu(w^i) \leq \frac{1}{k} \sum_{i=1}^{k} \left\| \frac{1}{\nu}A(x^{i-1} - x^i) \right\|^2 \leq \frac{2}{\nu k}[p_\nu(w^0) - p_\nu^*].
\]

We thus obtain a sublinear rate of convergence for the optimality condition. Note that this rate is independent of linear map \(A\).
3.2.2 Convex Case

**Assumption 2** $h$ and $g$ are both proper closed convex functions.

In this case, $h(w) + g_\nu(w)$ is a sum of a convex nonsmooth and convex smooth functions. This problem class has been exhaustively studied; see e.g. the survey of Parikh et al. (2014). The FISTA algorithm (Beck and Teboulle, 2009), detailed in Algorithm 2, can achieve faster convergence rates for this problem than Algorithm 1.

**Theorem 5** If Assumption 2 holds, the iterates generated by Algorithm 1 satisfy,

$$p(w^k) - p^* \leq \frac{\|w^0 - w^*\|^2}{2\nu(k + 1)}.$$

**Algorithm 2** FISTA for $h(w) + g_\nu(w)$

**Input:** $w^0$

- Initialize: $k = 0$, $a_0 = 1$, $v^0 = w^0$

1: **while** not converge **do**

2: $w^{k+1} \leftarrow \text{prox}_{\nu h}(v^k - \nu \nabla g_\nu(v^k))$

3: $a^{k+1} \leftarrow \frac{1 + \sqrt{1 + 4a^k}}{2}$

4: $v^{k+1} \leftarrow w^{k+1} + \frac{a^k - a^{k+1}}{a^{k+1}}(w^{k+1} - w^k)$

5: $k \leftarrow k + 1$

6: **end while**

**Output:** $w^k$

**Theorem 6** If Assumption 2 holds, the iterates generated by Algorithm 2 satisfy (Beck and Teboulle, 2009):

$$p_\nu(w^k) - p^*_\nu \leq \frac{2\|w^0 - w^*\|^2}{\nu(k + 1)^2}.$$

3.2.3 Strongly Convex Case

In two of our motivating examples, we take $g = 0$. In this case, we have a closed form solution for (8),

$$g_\nu(w) = \frac{1}{2\nu}||(I - PA)w||^2,$$

where $P_A = A(A^TA)^\dagger A^T$, and $\dagger$ denotes the pseudo inverse.

**Assumption 3** $h$ is $\alpha$-strongly convex and $g = 0$.

**Theorem 7** When Assumption 3 holds, the iterates generated by Algorithm 1 satisfy,

$$\|w^{k+1} - w^*\|^2 \leq \frac{1}{1 + \alpha\nu}\|w^k - w^*\|^2.$$

That is, we obtain a linear convergence rate in this case.
3.2.4 Sharp Minima Case

The final assumption concerns sharp minima, see Al-Khayyal and Kyparisis (1991); Cromme (1978); Hettich (1983); Polyak (1979); Burke and Ferris (1993) and Figure 2.

**Definition 8** We say the minimizer $w^*$ of $p_\nu$ is a sharp minimum, if there exist $\delta, \alpha > 0$, such that,

$$p_\nu(w) - p_\nu(w^*) \geq \alpha \|w - w^*\|, \quad \forall w \in \{w : \|w - w^*\| \leq \delta\}.$$ 

![Figure 2: Local function values grow quickly away from a sharp minimum.](image)

**Assumption 4** $h$ is proper closed convex, $g = 0$ and $w^*$ is a sharp minimum of $p_\nu$.

**Theorem 9** If Assumption 4 holds, and there exists an iteration $K$ with that,

$$\|w^k - w^*\| \leq \delta$$

then for all $k \geq K$, iterates generated by Algorithm 1 satisfy

$$\|w^{k+1} - w^*\| \leq \min \left\{ \|w^k - w^*\|, \frac{1}{\alpha \nu \|w^k - w^*\|^2} \right\}.$$ 

A sharp minimum gives us a local quadratic convergence rate.

4. Trimmed Nonconvex-Composite Models

We apply an analogous relaxation technique to problem class (2), obtaining the extended problem

$$\min_{v, x, w} f'_\nu(x, w, v) := \sum_{i=1}^m v_i h_i(w_i) + g(x) + \frac{1}{2\nu} \|Ax - w\|^2, \quad \text{s.t.} \ v \in \triangle_r,$$

where each function $h_i$ is nonsmooth and nonconvex. We use the notation $H(w) = [h_1(w_1), \ldots, h_m(w_m)]^T$, so that $\sum_{i=1}^m v_i h_i(w_i) = \langle v, H(w) \rangle$. 


Just as in Section 3, we partially minimize in $x$, reducing (17) to problem

$$\min_{v,w} p^t_{\nu}(w,v) := \sum_{i=1}^m v_i h_i(w_i) + g_{\nu}(w), \quad \text{s.t. } v \in \Delta_\tau$$  \hspace{1cm} (18)

The structure of (17) suggests a coordinate-descent algorithm detailed in Algorithm 3.

The operator $\text{prox}_{\nu(\langle v,H \rangle)}$ decouples across coordinates; for each nonzero $v_i$, we have

$$\text{prox}_{\nu(\langle v,H \rangle)}(\bar{w}) = \begin{bmatrix} \argmin_{w_1} \frac{1}{2\nu v_1}(w_1 - \bar{w}_1)^2 + h_1(w_1) \\ \vdots \\ \argmin_{w_m} \frac{1}{2\nu v_m}(w_m - \bar{w}_m)^2 + h_m(w_m) \end{bmatrix}.$$  

We now develop a convergence analysis for Algorithm 3. Our goal is to find the stationary point of (18), defined as follows.

**Definition 10** We call the pair $(\bar{w}, \bar{v})$ a stationary point of (18) when

$$0 \in \begin{bmatrix} \bar{v}_1 \partial h_1(\bar{w}_1) \\ \vdots \\ \bar{v}_m \partial h_m(\bar{w}_m) \end{bmatrix} + \nabla g_{\nu}(\bar{w}) := S^t_{\nu}(\bar{w}, \bar{v}), \quad 0 \in H(\bar{w}) + \partial \delta(\bar{v}|\Delta_\tau) := S^{\delta}_{\nu}(\bar{w}, \bar{v}).$$

We define the following quantity to characterize stationarity:

$$T^t_{\nu}(w,v) = \min \left\{ \frac{\nu}{2} \|s\|^2 + \alpha \|r\|^2 : s \in S^t_{\nu}(w,v), r \in S^{\delta}_{\nu}(w,v) \right\}.$$  

The convergence result is detailed in Theorem 11.

**Theorem 11** Denote by $w^k$ and $v^k$ the iterates generated by Algorithm 3. We have the following inequality,

$$T^t_{\nu}(w^{k+1},v^{k+1}) \leq p^t_{\nu}(w^k,v^k) - p^t_{\nu}(w^{k+1},v^{k+1}).$$

Moreover, by manipulating this inequality we obtain

$$\frac{1}{k} \sum_{i=1}^k T^t_{\nu}(w^i,v^i) \leq \frac{1}{k} [p^t_{\nu}(w^0,v^0) - p^t_{\nu}(w^k,v^k)],$$

which gives a sublinear rate of convergence for Algorithm 3.

**Proof** The proof is given in Appendix A. \[ \Box \]
5. Numerical Comparisons, Continuation, and Inexact Strategies.

In this section we provide numerical experiments that help to better understand Algorithm 1. In Section 5.1, we compare with the Alternating Directions Method of Multipliers (ADMM) in the convex setting. The iterations of ADMM are similar to those of Algorithm 1, with the augmented Lagrangian parameter $\rho$ in ADMM analogous to the relaxation parameter $\nu$ for RS. However, ADMM performs worse than RS in a direct comparison: it needs a larger number of iterations to achieve a specified error tolerance across choices of $\rho$ and $\nu$, and RS can achieve better practical performance, depending on the application. In Section 5.2, we discuss continuation strategies in $\nu$, that become important when RS is used iteratively to approximate the original problem (1). Finally, in Section 5.3 we consider large-scale problems where problem (8) cannot be solved in closed form, and iterative methods are required.

5.1 Comparison to ADMM in the Convex Setting

Although Algorithm 1 bears a strong resemblance to the ADMM algorithm (Algorithm 4, see e.g. Boyd et al. (2011)), they are fundamentally different:

- ADMM is a primal-dual method solving (1) while Algorithm 1 is a primal-only approach for solving the relaxation (7).

- ADMM has convergence guarantees for convex objectives $^2$, while Algorithm 1 is provably convergent both convex and nonconvex optimization problems.

**Algorithm 4** ADMM for convex $h(Ax) + g(x)$

**Input:** $x^0, \rho, \alpha$

**Initialize:** $k = 0, w^0, u^0$

1: while not converge do
2: $x^{k+1} \leftarrow \arg\min_x g(x) + \langle u^k, Ax - w^k \rangle + \frac{\rho}{2} \|Ax - w^k\|^2$
3: $w^{k+1} \leftarrow \text{prox}_{\frac{h}{\rho}}(Ax^{k+1} - u^k/\rho)$
4: $u^{k+1} \leftarrow u^k - \alpha(Ax^{k+1} - w^{k+1})$
5: $k \leftarrow k + 1$
6: end while

**Output:** $w^k$

We compare the two algorithms on a simple objective.

**Example 5** Consider $\ell_1$ linear regression,

$$
\min_x \|Ax - b\|_1. \tag{19}
$$

The quadratic relaxation (1) is given by

$$
\min_{x, w} \|w - b\|_1 + \frac{1}{2\nu} \|Ax - w\|^2. \tag{20}
$$

---

2. Convergence for nonconvex problems requires additional assumptions, see e.g. Wang et al. (2015)
Here $A \in \mathbb{R}^{m \times n}$ and $x_t \in \mathbb{R}^n$ are generated from standard Gaussian distribution, and $b = Ax_t + \epsilon + o$ with $\epsilon$ to be random Gaussian noise, and $o$ to be sparse outliers. We denote the solution to (19) as $x_{t_1}$ and the solution to (20) as $x_\nu$.

![Figure 3: Comparison between Algorithm 1 and ADMM. Left: number of iterations required by ADMM (blue) and Algorithm 1 (orange) to converge to a fixed tolerance, as a function of varying $\rho = 1/\nu$. Right: relative error of the solution obtained from ADMM (blue) and Algorithm 1 (orange) with respect to $x_t$.]

The numerical results are shown in Figure 3. In the experiments, we fix the augmented Lagrangian coefficient $\rho$ in ADMM to be equal to $1/\nu$, and this quantity from 1 to 100. We then plot the number of iterations required to hit a specified error tolerance, as well as the relative error of the recovered solution with respect to $x_t$.

As shown in the left plot of Figure 3, the number of iterations of Algorithm 1 grows linearly as a function of $1/\nu$, but is always below the number required by ADMM. The right figure of Figure 3 tells an interesting story. The relaxation may be more accurate than the original problem, depending on the application. When $\rho = 1/\nu = 10$, the solution of the relaxed formulation (20) is closer to the true model that that of (19), and Algorithm 1 can solve (20) much faster than ADMM can solve (19). Both the improvement in accuracy and the computational advantage persist as $\nu \downarrow 0$. In this problem, ADMM and RS iterations have exactly the same complexity, so the iterations comparison tells the full story.

5.2 Continuation

In the previous section, the solution obtained from the relaxed objective was closer to the true model. In other cases, such as noiseless phase retrieval, (7) and (1) can share a minimizer at a large value of $\nu$. However, more generally we may want to use (7) as an approximation to (1), in which case we want to explore continuation schemes with $\nu \downarrow 0$.

**Theorem 12** If $h$ is $L$-Lipschitz continuous and $(\bar{x}, \bar{w})$ is a stationary point of (7), we have,

$$
\|A\bar{x} - \bar{w}\| \leq L\nu.
$$

Moreover, when $A\bar{x} = \bar{w}$, we know $\bar{x}$ is also a stationary point of (1).
From Theorem 12 we know that, as $\nu$ goes to 0, solutions of (7) approach the solution set of (1). This yields a simple continuation strategy. Using the setting of Example 5, we take a decreasing positive sequence $\{\nu^k\}$, and initialize $x_{\nu^{k-1}}$ at the previous solution $x_{\nu^{k-1}}$.

We generate $A$ at different dimensions $m \in \{500, 1000, 2000, 5000, 10000\}$ with $n = 200$, and compare the results from the continuation strategy of Algorithm 1 continuation with the Julia Convex Package (which uses the splitting cone solver (SCS)). We check the final objective for (19), as well as the run times. Results are shown at Figure 4.

![Figure 4](image.png)

**Figure 4:** Comparison between Algorithm 1 continuation and the Julia Convex package with SCS. Left: objective values for (19) Algorithm 1 (blue) and SCS (green) as a function of $m$; the continuation approach finds the same or lower objective value as SCS. Right: run times for Algorithm 1 (blue) with SCS (green) as a function of $m$. The total work of the continuation approach is far less than required by SCS as $m$ increases.

Algorithm 1 gets a slightly lower objective value than the SCS algorithm; it is also far faster in terms of run-time, as shown in Figure 4. We emphasize that here SCS and Algorithm 1 are solving the same objective (19), since we drive $\nu \downarrow 0$ using a continuation strategy.

### 5.3 Inexact Solutions

Each iteration of Algorithm 1 requires solving a linear system. The potential drawback of Algorithm 1 is the computational cost for problem (8), especially for large scale problems. In many imaging applications, $A$ is an orthogonal operator, like the Fourier transform, Wavelet transform or Hadamard matrix; as a result, problem (8) in Algorithm 1 is tractable at acale. In more general applications, when the matrix $A$ is of moderate size, $A^\top A + \frac{1}{\nu} I$ can be pre-factored, and the factors used to solve (8). However, for large-scale systems $A$ may only be accessible through matrix-vector multiplication, and inexact solves of (8) are required to make Algorithm 1 practical.

Again using the setup in Example 5, we consider iterative methods, including preconditioned CG (Hestenes and Stiefel, 1952) and LSQR (Paige and Saunders, 1982) to solve the problem for large $n$. 

15
| CondNum | Alg. 1 iters | Total BFGS | Alg. 1 time(s) |
|---------|--------------|------------|----------------|
| 1       | 12           | 12         | 0.74           |
| 10      | 15           | 1099       | 18.28          |
| 20      | 20           | 1040       | 18.65          |
| 50      | 35           | 1054       | 22.87          |
| 100     | 60           | 1104       | 32.28          |

Table 3: Iterations and run times for Alg. 1 with BFGS solving (8). As the condition number grows, the total number of BFGS iterations used by Alg. 1 stays bounded.

In this experiment, we choose $m = 5000$, $d = 1000$, $\nu = 1$ and generate random matrices $A$ with different condition numbers. We use BFGS (see e.g. (Fletcher, 2013)) as the inner solver for (8). As the condition number increases, Algorithm 1 behaves quite well in the large-scale setting, as the total number of inner iterations stays bounded.

6. Machine Learning Applications

In this section, we give more detailed explanations for the motivating examples, and present numerical experiments and results. Phase Retrieval and its trimmed variant is presented in Section 6.1. Semi-supervised classification is considered in Section 6.2. The stochastic shortest path problem is developed in Section 6.3. New approaches for convex and nonconvex clustering are discussed in Section 6.4.

6.1 Sharp Phase Retrieval

Phase retrieval was originally introduced in signal processing for the X-ray crystallography problem Harrison (1993); Millane (1990) and arises in such diverse fields as microscopy (Miao et al., 1999; Frank et al., 2000; Drenth et al., 1975), holography (Fienup, 1980; Szöke, 1997), neutron radiography (Allman et al., 2000), optical design (Farn, 1991), adaptive optics, and astronomy. For a detailed review of applications and algorithms, see the survey of Luke et al. (2002).

Many algorithms has been studied by Fienup (1978, 1982); Gerchberg (1972). Recently, phase retrieval has gained some attention with the work of Candes et al. (2015); Duchi and Ruan (2017c); Eldar and Mendelson (2014) and Davis et al. (2017).

We consider an exact formulation of phase retrieval problem,

$$
\min_x \| |Ax| - b \|_1
$$

(21)

where $x$ is the signal we want to recover, $| \cdot |$ is the modulus of a complex number, and $b$ are the observed moduli obtained from linear observations $A$ of the true signal. We take $h_i(z) = |z| - b_i$, $g(x) = 0$ and optimize

$$
\min_{x, w} \| |w| - b \|_1 + \frac{1}{2\nu} \| Ax - w \|^2.
$$

(22)

We assume there is no noise in the experiment, so that $b = |Ax^*|$. In this case, (22) and (21) share the same solution.
We test Algorithm 1 on a large scale phase retrieval problem. We use a color image that is $2048 \times 2048$, with $m = 9 \times 2^{22}$ observations and $n = 3 \times 2^{22}$ unknowns. We define $H_n$ to be a normalized Walsh-Hadamard transform:

$$H_n \in \{-1, 1\}^{n \times n} / \sqrt{n}, \quad H_n = H_n^\top, \quad H_n^2 = I.$$ 

The linear operator $A$ is given by

$$A = \begin{bmatrix} H_n S_1 \\ \vdots \\ H_n S_k \end{bmatrix} \in \mathbb{R}^{kn \times n},$$

with $k = 4$ and $S_1, \ldots, S_k \in \text{diag}(-1, 1)^n$.

![Figure 5: Convergence history for large-scale phase retrieval.](http://getwallpapers.com/wallpaper/full/8/5/0/651422.jpg)

![Figure 6: Large example ($d = 3 \times 2^{22}, m = 3 \times 2^{22}$). Original picture (left), initial point (middle), and final result (right).](http://getwallpapers.com/wallpaper/full/8/5/0/651422.jpg)

The results are shown in Figures 5 and 6. The initialization algorithm works well, and Algorithm 1 converges within 30 iterations. Even though hypotheses of Theorem 9 do not

---

3. [http://getwallpapers.com/wallpaper/full/8/5/0/651422.jpg](http://getwallpapers.com/wallpaper/full/8/5/0/651422.jpg)
hold ($h$ is nonconvex), we expect a local quadratic rate of convergence since the minimum is sharp, and we observe this rate Figure 5.

**Comparison to State-of-the-Art Phase Retrieval Algorithms.** We compare Algorithm 1 with other methods developed by Duchi and Ruan (2017c) and by Davis et al. (2017). We summarize the results in Table 4.

| algorithm | objective | picture size | dimension | # meas | # FHT |
|-----------|-----------|--------------|-----------|--------|-------|
| Algorithm 1 | $\|Ax\|_1 - b\|_1$ | 2048$^2$ | $n = 3 \times 2^{22}$ | $m = 3n$ | 518 $^*$ |
| Duchi and Ruan (2017c) | $\|(Ax)^2 - b\|_1$ | 1024$^2$ | $n = 3 \times 2^{20}$ | $m = 3n$ | 15100 $^*$ |
| Davis et al. (2017) | $\|(Ax)^2 - b\|_1$ | 2048$^2$ | $n = 3 \times 2^{22}$ | $m = 3n$ | 1530 $^*$ |

Table 4: Comparison summary. FHT stands for fast Hadamard transform. The number of FHTs include those used during initialization.

Algorithm 1 uses fewer matrix vector multiplications (fast Hadamard transforms) to obtain the solution, compared to recently developed phase retrieval algorithms. The counts include initialization, with Algorithm 1 using 10 power iterations to initialize, while Davis et al. (2017) start at random point. For this problem, Algorithm 1 is minimizing a different objective than the other methods, see Table 4. However, we can compare the Hadamard counts directly since all methods recover the true phase.

**Trimmed Phase Retrieval.** The measurements of the magnitude can be corrupted due to detector malfunction, heteroscedastic noise, or physical limitations. A robust extension of phase retrieval is needed in these situations. We use the trimmed extension of (21):

$$\min_{v,x} \sum_{i=1}^m v_i |\langle a_i, x \rangle - b_i|, \quad \text{s.t. } v \in \triangle \tau,$$

where $\tau$ indicates the estimated number of good measurements. This is a nonsmooth trimming problem, and we use TRS, see Section 4. The relaxed trimmed phase retrieval objective is given by

$$\min_{w,v,x} \frac{1}{2} \sum_{i=1}^m v_i |w_i - b_i|^2 + \frac{1}{2\nu} \|Ax - w\|^2, \quad \text{s.t. } v \in \triangle \tau.$$

In the experiments, we use a small MNIST$^4$ picture as the data source with dimension $n = 28 \times 28 = 784$. We take $m = 5n$, measurements, and corrupt 30% of them by replacing the measurements with large scalar 1000. We then solve both (22) and (24). Trimming makes a significant difference in the quality of the recovered image, see Figure 7.

### 6.2 Semi-Supervised Classification

Classification is a fundamental problem in machine learning. Logistic Regression (McCullagh and Nelder (1989)) and Support Vector Machines (SVMs, see Cortes and Vapnik

4. [http://yann.lecun.com/exdb/mnist/](http://yann.lecun.com/exdb/mnist/)
(1995)) are used widely for binary classification; training requires labeled examples. In many applications, labeling the data can be a slow, costly and error-prone process. Semi-supervised learning attempts to use both labeled and unlabeled data to improve accuracy (relative to using only labeled data).

Logistic regression for binary classification is both easily formulated and widely used. We consider the semi-supervised logistic regression (SSLR).

Building on early work for semi-supervised classification in the pattern recognition community (see survey in McLachlan (2004)), Amini and Gallinari (2002) proposed a variant of SSLR, building a discriminant logistic model and using a Classification Expectation Maximization (CEM) algorithm to solve the resulting formulation. The work of Amini and Gallinari (2002) and follow-up papers (e.g. Madani et al. (2005)) share a key theme: they estimate posterior probabilities of class labels, which are then used in the maximization step. The idea of taking expectations over class labels brings the Expectation-Maximization (EM) algorithm to bear on the model.

Our approach to semi-supervised logistic regression is inspired by transductive SVMs, introduced by Vapnik and Sterin (1977). The more modern variant of the problem is often called the semi-supervised SVM (S3VM), see e.g. work of Chapelle et al. (2008).

Following the intuition of transductive SVMs, we want to solve the logistic regression problem while separating unlabeled data as well as possible, regardless of the label. This leads to an intuitively simple nonsmooth, nonconvex problem

$$\min_x \sum_{i=1}^{l} \log(1 + \exp(-b_i \langle a_i, x \rangle)) + \gamma \sum_{i=l+1}^{m} \log(1 + \exp(-|\langle a_i, x \rangle|)) + \frac{\lambda}{2} \|x\|^2,$$

(25)

where $a_i \in \mathbb{R}^n$ is the data image, $b_i \in \{-1, 1\}$ is the label and $\gamma$ controls the weight of the semi-supervised part. Without loss of generality, we assume only the first $l$ images are labeled. Geometrically, when data is labeled, the direction to push the classifier is determined; when data is unlabeled, we tend to push the classifier in both ways depend on its current position.

Problem (25) is different from all previous SSLR formulations, and in particular does not require an EM algorithm; it can be optimized directly. Problem (25) falls squarely into

Figure 7: The advantages of trimming phase retrieval: (a) is the true data source, (b) is the initial starting point, (c) phase retrieval results using (22), (d) trimmed phase retrieval results using (24).
the framework we proposed in this paper, and the relaxed objective can be written as,

$$\min_{x,w} \sum_{i=1}^{l} \log(1 + \exp(-b_i w_i)) + \gamma \sum_{i=l+1}^{m} \log(1 + \exp(-|w_i|)) + \frac{1}{2\nu} \|Ax - w\|^2 + \frac{\lambda}{2} \|x\|^2. \quad (26)$$

If we treat (26) as a specification of (7) we have,

$$g(x) = \frac{\lambda}{2} \|x\|^2, \quad h_i(z) = \begin{cases} \log(1 + \exp(-b_i z)), & i \leq l \\ \log(1 + \exp(-|z|)), & i > l \end{cases},$$

and when $i > l$ we know that $h_i$ is nonconvex and nonsmooth.

To apply Algorithm 1, closed form solution of (8) can be obtained. We also need to calculate the proximal operator of $h_i$. For $i \leq l$, the prox-subproblems is smooth and convex. For $i > l$, i.e. for the unlabeled examples, the prox problem in each coordinate requires solving the scalar problem,

$$\min_{w_i} \frac{1}{2\nu} (w_i - \overline{w}_i)^2 + \gamma \log(1 + \exp(-|w_i|)).$$

The optimal $z$ will necessarily have the same sign as $\overline{z}$, and so we can rewrite the problem

$$\min_{|w_i|} \frac{1}{2\nu} (|w_i| - |\overline{w}_i|)^2 + \gamma \log(1 + \exp(-|w_i|)).$$

This is again a smooth and convex problem in $w$, so we can apply Newton’s method to find $|\tilde{w}_i|$. The solution $\tilde{w}_i$ is then immediately obtained by $\tilde{w}_i = |\tilde{w}_i| \text{sign}(\overline{w}_i)$.

Our goal in the experimental results is to illustrate the simplicity and flexibility of the new SSLR concept. We leave a comprehensive comparison with prior art on semi-supervised classification to future work.

Figure 8 shows the convergence result for run of the algorithm, with parameters $m = 12665$, $l = 254$ (2% of data labeled), $\lambda = 0.1$, $\gamma = 0.1$ and $\nu = 1$. Consistently with Theorem 5, when $h$ is nonconvex, Algorithm 1 has a sublinear rate.

To evaluate the results, we focus on prediction accuracy as a function of the $\gamma$ parameter in (26), and fix $\lambda = 0.1$, $\nu = 1$. We let $\gamma$ range among $0, 0.1, \ldots, 0.9, 1$. We use two sets of MNIST data, considering binary classification of digit pairs (0, 1) and (4, 9). For each choice of $\gamma$, we conduct 20 random trails and record the mean and variance of the test accuracy.

Testing errors are shown in Figure 9. Several observation can be made.

• (4, 9) yields a harder classification problem compared with (0, 1). For each ratio of labeled to unlabeled data, test accuracy for (4, 9) is lower than for (0, 1).

• Semi-supervised learning helps more for the MNIST dataset when we have very few labeled datapoints.

• The variance of accuracy results increases with $\gamma$ (as we pay more attention to unlabeled data), and decrease with ratio of labeled to unlabeled data.
Figure 8: Convergence plot for semi-supervised Logistic Regression.

Figure 9: Testing errors of semi-supervised logistic regression. Left: results of the (0, 1) classification experiment. Right: results of the (4, 9) classification experiment. Both plots show the test errors as a function $\gamma$, with 2% labeled data (blue) and 5% labeled data (orange). The dotted lines and colored areas show the mean and range the results obtained across 20 random trails.

We see the lowest test error for $\gamma = 0.1$ across all experiments.

The results show that some degree of improvement is readily obtained from the SSLR strategy, and that the proposed approach can easily handle the new type of optimization problem. We leave extensions to more powerful learning models and comparisons with the robust literature on semi-supervised classification to future work.

6.3 Stochastic Shortest Path

In this experiment, we consider the stochastic shortest path problem described by Bertsekas and Tsitsiklis (1991). For a review of the history of shortest path problem, please check Schrijver (2012). As shown in Figure 11, the version we consider looks for the minimum
The expected cost path from node A to node B, given a certain graph structure. At each node, we select between two graphs, then take a step by uniformly sampling available paths of the chosen graph to move to an adjacent node, paying the specified cost.

The specific example we consider contains \( n = 25 \) nodes. Two graphs are generated randomly, along with the cost matrices \( C^1, C^2 \in \mathbb{R}^{n \times n} \) for each graph, with \( C^k_{ij} \) defined as the cost \(^5\) to move from node \( i \) to node \( j \) within graph \( k \). We also let \( U^1, U^2 \in \mathbb{R}^{n \times n} \) denote the connectivity matrices, with entry \( U^k_{ij} \) encoding the probability that node \( i \) moves to node \( j \) within graph \( k \).

If we set \( x^* \in \mathbb{R}^n \) as the optimal cost with the \( i \)-th entry representing best expected cost starting from node \( i \), we use the Bellman equation (see Bellman (1958))

\[
x^*_i = \min \{ \mathbb{E}[C^1_{ij} + x^*_j], \mathbb{E}[C^2_{ij} + x^*_j] \} = \min \{ \langle u^1_i, c^1_i + x^* \rangle, \langle u^2_i, c^2_i + x^* \rangle \}
\]

and to formulate the stochastic shortest path as a deterministic optimization problem:

\[
\min_x \sum_{i=1}^d |x_i - \min \{ \langle u^1_i, x \rangle + v^1_i, \langle u^2_i, x \rangle + v^2_i \}|
\]  
(27)

where \( u^k_i \) is the \( i \)-th row of \( U^k \) and \( v^k_i = \langle u^k_i, c^k_i \rangle \) with \( c^k_i \) the \( i \)th row of \( C^k \) for \( k = 1, 2 \).

Problem (27) is nonsmooth and nonconvex; and using the method in the manuscript we write the approximate problem

\[
\min_{x, w^1, w^2} h(w^1, w^2) + \frac{1}{2\nu} \left( \|A^1 x - w^1\|^2 + \|A^2 x - w^2\|^2 \right)
\]  
(28)

where \( A^k = U^k - I \), and \( h(w^1, w^2) = \sum_{i=1}^d \min \{ w^1_i + v^1_i, w^2_i + v^2_i \} \).

The optimal value of (28) is 0 because there is a solution to the Bellman equation. For the same reason, the solution of (27) and (28) coincide. The convergence results are shown in Figure 10, where we see a linear convergence rate in Figure 10. The obtained optimal policy is shown in Figure 11.

![Figure 10: Convergence plot of stochastic shortest path experiment.](image-url)
6.4 Convex and Nonconvex Clustering Problem

Clustering is a fundamental unsupervised learning technique. Basic approaches including $k$-means (Hartigan and Wong, 1979) and mixture models (Dempster et al., 1977) are popular due to their simplicity and statistical interpretation. These approaches are built on essentially combinatorial subproblems (e.g. assigning members to clusters), making the approaches vulnerable to stalling at local minima. More recently, convex clustering formulations were proposed by Lindsten et al. (2011) and Hocking et al. (2011).

The recent clustering formulations take the form

$$\min_X \frac{1}{2} \sum_{i=1}^{m} \|x_i - u_i\|^2 + \lambda \sum_{i=1}^{m-1} \sum_{j=i+1}^{m} \rho(x_i - x_j), \quad (29)$$

where $U = [u_1, \ldots, u_m]$ are the data points, $X = [x_1, \ldots, x_m]$ are the decision variables and $\rho$ is the fusion regularizer. In the convex setting, $\rho$ usually is chosen as the $\ell_2$ norm, to encourage $x_i = x_j$; the number of different elements is controlled by the penalty $\lambda$. Problem (29) is then solved using splitting methods, including ADMM 4, or the alternating minimization algorithm (AMA) as proposed by Chi and Lange (2015). The proposed RS approach is a natural competitor, especially given the results of Section 5.1.
Relaxing problem (29), we get the objective

\[
\min_{x,w} \frac{1}{2} \sum_{i=1}^{m} \|x_i - u_i\|^2 + \lambda \sum_{i=1}^{m-1} \sum_{j=i+1}^{m} \rho(w_{ij}) + \frac{1}{2\nu} \sum_{i=1}^{m-1} \sum_{j=i+1}^{m} \|x_i - x_j - w_{ij}\|^2.
\]  

(30)

Algorithm (1) requires only a regularized least squares solve, and the proximal operator for \(\rho\); it can be applied to both convex and nonconvex fusion penalties.

**Comparison with ADMM.** In this experiment, we generate a synthetic data set, with three clusters and 10 points per cluster. The hyper parameters are chosen as \(\lambda = 0.5\) and \(\nu = 1\). Results are shown in Figure 12, where we compare with ADMM and show the final adjacency matrix obtained from \(w_{ij}\).

From the right plot of Figure 12, we can see that convex clustering via (29) and (30) cleanly identifies the clusters with these parameters. The left plot of Figure 12 shows identical performance between Algorithms 1 for (30) (blue) and ADMM for (29) (beige).

De-Biased Clustering. One issue with (30) is that \(\rho = \|d\|_2\) is very sensitive to \(\lambda\), because of the bias introduced by points from different clusters. For this specific reason, we consider a nonconvex SCAD (Fan and Li, 2001)-like regularizer,

\[
\rho(d;\kappa) = \begin{cases} 
\|d\|, & \|d\| \leq \kappa \\
0, & \|d\| > \kappa \end{cases}
\]

This regularizer allows us to use prior knowledge on the radius of each cluster, encoded by \(\kappa\). This prior knowledge makes tuning \(\lambda\) easier, and also speeds up convergence of the clustering.
Nonconvex Splitting Methods

There is no convergence guarantee for ADMM when the SCAD penalty is used; however it still converges, even faster than for the convex case. Algorithm (1) is guaranteed to converge for (30), and has a significantly faster rate, see the left plot of Figure 12.

To test behavior with respect to the fusion penalty \( \lambda \), we allow \( \lambda \) to vary in a grid from 0 to 1, and plot the path of the variables \( x_i \). We also compare the convergence results for \( \lambda = 0.5 \) between convex and nonconvex \( \rho \). These results are shown in Figure 13.

When we use clustering fusion penalties, all points affect one another; for larger values of the penalty \( \lambda \), all points are rapidly assigned to a single cluster with center given by the center of mass of the point cloud. In contrast, using the nonconvex SCAD allows clusters that are far enough away to not affect each other, allowing desirable clustering behavior locally without the overall global effect.

![Comparison of the clustering paths for convex vs. nonconvex \( \rho \) across penalty parameters. Left: clustering path with convex \( \rho = \| \cdot \|_2 \). Right: clustering path of the variables using the nonconvex SCAD penalty \( \rho \). Nonconvex fusion penalties give additional modeling flexibility and interpretable results.](image)

7. Discussion

We have developed a new ‘relax and split’ approach for nonconvex-composite problems, and extended it to trimmed robust formulations. The approach applies to highly nonconvex models (those that are not even weakly convex), and can be easily applied to difficult structured nonsmooth nonconvex problems. The problem class is more general than those analyzed by recent sub-gradient based methods for nonsmooth nonconvex optimization.

We have also shown how the model and associated algorithms can be used for a variety of applications, including exact phase retrieval, semi-supervised classification, stochastic shortest path problems, and new approaches to clustering. Every such application can be ‘robustified’ with the trimming extension, as we showed using the outlier-contaminated phase retrieval problem.

The paper opens several new avenues and raises important questions for future work, including a comprehensive analysis of inexact ‘relax-and-split’ approaches, extensions to
compositions of nonconvex losses with nonlinear maps, and substantial detailed numerical work to evaluate the approach across a range of application domains.

**Acknowledgment.** Research of A. Aravkin was partially supported by the Washington Research Foundation Data Science Professorship.

**References**

F Al-Khayyal and J Kyparisis. Finite convergence of algorithms for nonlinear programs and variational inequalities. *Journal of Optimization Theory and Applications*, 70(2):319–332, 1991.

Andreas Alfons, Christophe Croux, Sarah Gelper, et al. Sparse least trimmed squares regression for analyzing high-dimensional large data sets. *The Annals of Applied Statistics*, 7(1):226–248, 2013.

BE Allman, PJ McMahon, KA Nugent, D Paganin, David L Jacobson, Muhammad Arif, and SA Werner. Imaging: phase radiography with neutrons. *Nature*, 408(6809):158, 2000.

Massih-Reza Amini and Patrick Gallinari. Semi-supervised logistic regression. In *ECAI*, pages 390–394, 2002.

Aleksandr Aravkin and Damek Davis. A smart stochastic algorithm for nonconvex optimization with applications to robust machine learning. *arXiv preprint arXiv:1610.01101*, 2016.

A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imaging Sci.*, 2(1):183–202, 2009. ISSN 1936-4954. doi: 10.1137/080716542. URL http://dx.doi.org/10.1137/080716542.

Stephen Becker, Jérôme Bobin, and Emmanuel J Candès. Nesta: A fast and accurate first-order method for sparse recovery. *SIAM Journal on Imaging Sciences*, 4(1):1–39, 2011.

Richard Bellman. On a routing problem. *Quarterly of applied mathematics*, 16(1):87–90, 1958.

Dimitri P Bertsekas and John N Tsitsiklis. An analysis of stochastic shortest path problems. *Mathematics of Operations Research*, 16(3):580–595, 1991.

Jérôme Bolte, Shoham Sabach, and Marc Teboulle. Proximal alternating linearized minimization for nonconvex and nonsmooth problems. *Mathematical Programming*, 146(1-2):459–494, 2014.

Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, Jonathan Eckstein, et al. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine learning*, 3(1):1–122, 2011.

J. V. Burke and M. C. Ferris. A gauss—newton method for convex composite optimization. *Mathematical Programming*, 71(2):179–194, 1995. ISSN 1436-4646. doi: 10.1007/BF01585997.
James V Burke. Descent methods for composite nondifferentiable optimization problems. *Mathematical Programming*, 33(3):260–279, 1985a. doi: 10.1007/BF01584377.

James V Burke, Adrian S Lewis, and Michael L Overton. A robust gradient sampling algorithm for nonsmooth, nonconvex optimization. *SIAM Journal on Optimization*, 15(3):751–779, 2005.

J.V. Burke. Descent methods for composite nondifferentiable optimization problems. *Math. Programming*, 33(3):260–279, 1985b. ISSN 0025-5610. doi: 10.1007/BF01584377. URL http://dx.doi.org/10.1007/BF01584377.

JV Burke and Michael C Ferris. Weak sharp minima in mathematical programming. *SIAM Journal on Control and Optimization*, 31(5):1340–1359, 1993.

Emmanuel J Candes, Yonina C Eldar, Thomas Strohmer, and Vladislav Voroninski. Phase retrieval via matrix completion. *SIAM review*, 57(2):225–251, 2015.

C. Cartis, N.I.M. Gould, and P.L. Toint. On the evaluation complexity of composite function minimization with applications to nonconvex nonlinear programming. *SIAM J. Optim.*, 21(4):1721–1739, 2011. ISSN 1052-6234. doi: 10.1137/11082381X.

Olivier Chapelle, Vikas Sindhwani, and Sathiya S Keerthi. Optimization techniques for semi-supervised support vector machines. *Journal of Machine Learning Research*, 9(Feb): 203–233, 2008.

Eric C Chi and Kenneth Lange. Splitting methods for convex clustering. *Journal of Computational and Graphical Statistics*, 24(4):994–1013, 2015.

Andrew R Conn, Katya Scheinberg, and Luis N Vicente. *Introduction to derivative-free optimization*. SIAM, 2009.

Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

Ludwig Cromme. Strong uniqueness. *Numerische Mathematik*, 29(2):179–193, 1978.

Damek Davis and Dmitriy Drusvyatskiy. Stochastic subgradient method converges at the rate $O(k^{-1/4})$ on weakly convex functions. *arXiv preprint arXiv:1802.02988*, 2018.

Damek Davis, Dmitriy Drusvyatskiy, and Courtney Paquette. The nonsmooth landscape of phase retrieval. *arXiv preprint arXiv:1711.03247*, 2017.

Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38, 1977.

AJJ Drenth, AMJ Huiser, and HA Ferwerda. The problem of phase retrieval in light and electron microscopy of strong objects. *Optica Acta: International Journal of Optics*, 22(7):615–628, 1975.
D. Drusvyatskiy and A.S. Lewis. Error bounds, quadratic growth, and linear convergence of proximal methods. *To appear in Math. Oper. Res.*, arXiv:1602.06661, 2016.

D. Drusvyatskiy and C. Paquette. Efficiency of minimizing compositions of convex functions and smooth maps. *Preprint arXiv:1605.00125*, 2016.

J.C. Duchi and F. Ruan. Stochastic methods for composite optimization problems. *Preprint arXiv:1703.08570*, 2017a.

J.C. Duchi and F. Ruan. Solving (most) of a set of quadratic equalities: Composite optimization for robust phase retrieval. *Preprint arXiv:1705.02356*, 2017b.

John C Duchi and Feng Ruan. Solving (most) of a set of quadratic equalities: Composite optimization for robust phase retrieval. *arXiv preprint arXiv:1705.02356*, 2017c.

Yonina C Eldar and Shahar Mendelson. Phase retrieval: Stability and recovery guarantees. *Applied and Computational Harmonic Analysis*, 36(3):473–494, 2014.

Jianqing Fan and Runze Li. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American statistical Association*, 96(456):1348–1360, 2001.

Michael W Farn. New iterative algorithm for the design of phase-only gratings. In *Computer and Optically Generated Holographic Optics; 4th in a Series*, volume 1555, pages 34–43. International Society for Optics and Photonics, 1991.

James R Fienup. Reconstruction of an object from the modulus of its fourier transform. *Optics letters*, 3(1):27–29, 1978.

James R Fienup. Phase retrieval algorithms: a comparison. *Applied optics*, 21(15):2758–2769, 1982.

JR Fienup. Iterative method applied to image reconstruction and to computer-generated holograms. *Optical Engineering*, 19(3):193297, 1980.

R. Fletcher. A model algorithm for composite nondifferentiable optimization problems. *Math. Programming Stud.*, (17):67–76, 1982. ISSN 0303-3929. Nondifferential and variational techniques in optimization (Lexington, Ky., 1980).

Roger Fletcher. *Practical methods of optimization*. John Wiley & Sons, 2013.

Joachim Frank, Pawel Penczek, Rajendra K Agrawal, Robert A Grassucci, and Amy B Heagle. [18] three-dimensional cryoelectron microscopy of ribosomes. 2000.

Ralph W Gerchberg. A practical algorithm for the determination of the phase from image and diffraction plane pictures. *Optik*, 35:237–246, 1972.

N. Gillis. Introduction to nonnegative matrix factorization. *SIAG/OPT Views and News*, 25(1):7–16, 2017.

Robert W Harrison. Phase problem in crystallography. *JOSA a*, 10(5):1046–1055, 1993.
John A Hartigan and Manchek A Wong. Algorithm as 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1):100–108, 1979.

Magnus Rudolph Hestenes and Eduard Stiefel. *Methods of conjugate gradients for solving linear systems*, volume 49. NBS, 1952.

Rainer Hettich. A review of numerical methods for semi-infinite optimization. In *Semi-infinite programming and applications*, pages 158–178. Springer, 1983.

Toby Dylan Hocking, Armand Joulin, Francis Bach, and Jean-Philippe Vert. Clusterpath an algorithm for clustering using convex fusion penalties. In *28th international conference on machine learning*, page 1, 2011.

Adrian S Lewis and Michael L Overton. Nonsmooth optimization via bfgs. *Submitted to SIAM J. Optimiz*, pages 1–35, 2009.

A.S. Lewis and S.J. Wright. A proximal method for composite minimization. *Math. Program.*, pages 1–46, 2015. doi: 10.1007/s10107-015-0943-9. URL http://dx.doi.org/10.1007/s10107-015-0943-9.

Fredrik Lindsten, Henrik Ohlsson, and Lennart Ljung. *Just relax and come clustering!: A convexification of k-means clustering*. Linköping University Electronic Press, 2011.

D Russell Luke, James V Burke, and Richard G Lyon. Optical wavefront reconstruction: Theory and numerical methods. *SIAM review*, 44(2):169–224, 2002.

Omid Madani, David M Pennock, and Gary W Flake. Co-validation: Using model disagreement on unlabeled data to validate classification algorithms. In *Advances in neural information processing systems*, pages 873–880, 2005.

Peter McCullagh and John A Nelder. *Generalized linear models*, volume 37. CRC press, 1989.

Geoffrey McLachlan. *Discriminant analysis and statistical pattern recognition*, volume 544. John Wiley & Sons, 2004.

Jianwei Miao, Pambos Charalambous, Janos Kirz, and David Sayre. Extending the methodology of x-ray crystallography to allow imaging of micrometre-sized non-crystalline specimens. *Nature*, 400(6742):342, 1999.

Rick P Millane. Phase retrieval in crystallography and optics. *JOSA A*, 7(3):394–411, 1990.

B.S. Mordukhovich. *Variational analysis and generalized differentiation. I*, volume 330 of *Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences]*. Springer-Verlag, Berlin, 2006. ISBN 978-3-540-25437-9; 3-540-25437-4. Basic theory.

Yu Nesterov. Smooth minimization of non-smooth functions. *Mathematical programming*, 103(1):127–152, 2005.
Yu. Nesterov. Modified Gauss-Newton scheme with worst case guarantees for global performance. *Optim. Methods Softw.*, 22(3):469–483, 2007. ISSN 1055-6788. doi: 10.1080/08927020600643812. URL http://dx.doi.org/10.1080/08927020600643812.

J. Nocedal and S.J. Wright. *Numerical optimization*. Springer Series in Operations Research and Financial Engineering, Springer, New York, second edition, 2006. ISBN 978-0387-30303-1; 0-387-30303-0.

Christopher C Paige and Michael A Saunders. Lsqr: An algorithm for sparse linear equations and sparse least squares. *ACM transactions on mathematical software*, 8(1):43–71, 1982.

Neal Parikh, Stephen Boyd, et al. Proximal algorithms. *Foundations and Trends® in Optimization*, 1(3):127–239, 2014.

BT Polyak. Sharp minima. Institute of control sciences lecture notes, Moscow, USSR, 1979. In *IIASA workshop on generalized Lagrangians and their applications*, IIASA, Laxenburg, Austria, 1979.

M.J.D. Powell. General algorithms for discrete nonlinear approximation calculations. In *Approximation theory, IV (College Station, Tex., 1983)*, pages 187–218. Academic Press, New York, 1983.

M.J.D. Powell. On the global convergence of trust region algorithms for unconstrained minimization. *Math. Programming*, 29(3):297–303, 1984. ISSN 0025-5610. doi: 10.1007/BF02591998. URL http://dx.doi.org/10.1007/BF02591998.

R. T. Rockafellar. *Convex analysis*. Princeton Mathematical Series, No. 28. Princeton University Press, Princeton, N.J., 1970.

R.T. Rockafellar and R.J-B. Wets. *Variational Analysis*. Grundlehren der mathematischen Wissenschaften, Vol 317, Springer, Berlin, 1998.

Peter J Rousseeuw. Multivariate Estimation with High Breakdown Point. *Mathematical statistics and applications*, 8:283–297, 1985.

Alexander Schrijver. On the history of the shortest path problem. *Doc. Math*, 155, 2012.

Abraham Szöke. Holographic microscopy with a complicated reference. *Journal of Imaging Science and Technology*, 41(4):332–341, 1997.

V Vapnik and A Sterin. On structural risk minimization or overall risk in a problem of pattern recognition. *Automation and Remote Control*, 10(3):1495–1503, 1977.

Yu Wang, Wotao Yin, and Jinshan Zeng. Global convergence of admm in nonconvex nonsmooth optimization. *Journal of Scientific Computing*, pages 1–35, 2015.

S.J. Wright. Convergence of an inexact algorithm for composite nonsmooth optimization. *IMA J. Numer. Anal.*, 10(3):299–321, 1990. ISSN 0272-4979. doi: 10.1093/imanum/10.3.299. URL http://dx.doi.org/10.1093/imanum/10.3.299.
Mengwei Xu, J Ye Jane, and Liwei Zhang. Smoothing augmented lagrangian method for nonsmooth constrained optimization problems. *Journal of Global Optimization*, 62(4): 675–694, 2015.

E. Yang and A. Lozano. Robust Gaussian Graphical Modeling with the Trimmed Graphical Lasso. In *Advances in Neural Information Processing Systems*, pages 2602–2610, 2015.

E. Yang, A. Lozano, and A. Aravkin. High-Dimensional Trimmed Estimators: A General Framework for Robust Structured Estimation. *arXiv preprint arXiv:1605.08299*, 2016.

Junfeng Yang and Yin Zhang. Alternating direction algorithms for ℓ1-problems in compressive sensing. *SIAM journal on scientific computing*, 33(1):250–278, 2011.

Y. Yuan. On the superlinear convergence of a trust region algorithm for nonsmooth optimization. *Math. Programming*, 31(3):269–285, 1985. ISSN 0025-5610. doi: 10.1007/BF02591949. URL http://dx.doi.org/10.1007/BF02591949.
Appendix A. Proofs of Section 3

Proof [Theorem 1]

Observe that,

\[
\min_x g(x) + \frac{1}{2\nu} \|Ax - w\|^2 = \min_{x,y} \left\{ g(x) + \frac{1}{2\nu} \|y - w\|^2 : y = Ax \right\}
\]

Define \( Ag(y) = \min_x \{ g(x) : Ax = y \} \) which is the image of \( g \) under \( A \). From Rockafellar (1970) [Theorem 5.7] we know that \( Ag \) is a convex function. Moreover, since \( g \) is proper and bounded below, we know that \( Ag \) is also proper.

We cannot show \( Ag \) is closed unless we know more information about \( g \) and \( A \) Rockafellar (1970) [Theorem 9.2]. Instead we show that for every \( w \),

\[
g_{\nu}(w) = \tilde{g}_{\nu}(w) := \min_x (\text{cl} \ Ag)(y) + \frac{1}{2\nu} \|y - w\|^2,
\]

where \text{cl} denotes the closure of the function.

Since \((\text{cl} \ Ag)(y) \leq Ag(y)\) for all \( y \), we know that,

\[
g_{\nu}(w) \geq \tilde{g}_{\nu}(w).
\]

Since \((\text{cl} \ Ag)(y) + \frac{1}{2\nu} \|y - w\|^2\) is closed and strongly convex, we also know that there exist a unique minimizer,

\[
y^* = \text{argmin}_y (\text{cl} \ Ag)(y) + \frac{1}{2\nu} \|y - w\|^2.
\]

From Rockafellar (1970)[Theorem 7.5], we know, for some \( z \in \text{ri dom} \ Ag \),

\[
\text{cl} \ Ag(y^*) = \lim_{\lambda \uparrow 1} Ag(\lambda y^* + (1 - \lambda)z).
\]

Define the sequence \( \{y_\lambda\} \), such that, \( y_\lambda = \lambda y^* + (1 - \lambda)z \). Since \( y^* \in \text{dom cl} \ Ag = \text{cl dom} \ Ag \), using Rockafellar (1970)[Theorem 6.1] we know that for every \( 0 \leq \lambda < 1 \), \( y_\lambda \in \text{ri dom} \ Ag \). Therefore,

\[
\tilde{g}_{\nu}(w) = Ag(y^*) + \frac{1}{2\nu} \|y^* - w\|^2 = \lim_{\lambda \uparrow 1} Ag(y_\lambda) + \frac{1}{2\nu} \|y_\lambda - w\|^2 \geq g_{\nu}(w),
\]

so \( g_{\nu}(w) = \tilde{g}_{\nu}(w) \). From Rockafellar and Wets (1998) [Theorem 2.26] we know that \( g_{\nu}(w) \) is a closed convex function, with a \( \frac{1}{\nu} \)-Lipschitz continuous gradient,

\[
\nabla g_{\nu}(w) = \nabla \tilde{g}_{\nu}(w) = \frac{1}{\nu} (w - y^*).
\]

Since \( y^* \in \text{dom cl} \ Ag = \text{cl dom} \ Ag \subset \text{Range}(A) \), we define \( x^* = \{ x : Ax = y^* \} \). Then we have the desired result,

\[
\nabla g_{\nu}(w) = \frac{1}{\nu} (w - Ax), \quad \forall x \in x^*.
\]

\[\blacksquare\]
**Proof** (Theorem 4)
Using the iteration of Algorithm 1, and introducing the sequence \( \{x^k\} \), we have,
\[
0 \in \frac{1}{\nu} A^\top (Ax^k - w^k) + \partial g(x^k), \quad 0 \in \frac{1}{\nu} (w^k - Ax^k) + \partial h(w^k).
\]
From the definition of the objective, we have,
\[
p_{\nu}(w^k) = h(w^k) + \frac{1}{2\nu} \|Ax^k - w^k\|^2 + g(x^k)
\]
\[
= h(w^k) + \frac{1}{2\nu} \|Ax^{k-1} - w^k + A(x^k - x^{k-1})\|^2 + g(x^k)
\]
\[
\leq h(w^{k-1}) + \frac{1}{2\nu} \|Ax^{k-1} - w^k + A(x^k - x^{k-1})\|^2 + g(x^k)
\]
\[
+ \frac{1}{\nu} \langle Ax^{k-1} - w^k, A(x^k - x^{k-1}) \rangle
\]
\[
+ \frac{1}{\nu} \|Ax^{k-1} - w^k, A(x^k - x^{k-1}) \rangle + \frac{1}{2\nu} \|A(x^k - x^{k-1})\|^2 + g(x^k) - g(x^{k-1}).
\]
Since \( g \) is convex,
\[
g(x^k) - g(x^{k-1}) \leq \langle \partial g(x^k), x^k - x^{k-1} \rangle = \frac{1}{\nu} \langle w^k - Ax^k, A(x^k - x^{k-1}) \rangle.
\]
Therefore we have,
\[
p_{\nu}(w^k) - p_{\nu}(w^{k-1}) \leq \frac{1}{\nu} \langle Ax^{k-1} - w^k, A(x^k - x^{k-1}) \rangle + \frac{1}{2\nu} \|A(x^k - x^{k-1})\|^2
\]
\[
+ \frac{1}{\nu} \langle w^k - Ax^k, A(x^k - x^{k-1}) \rangle
\]
\[
= - \frac{1}{2\nu} \|A(x^{k-1} - x^k)\|^2.
\]
Summing up, we get,
\[
\frac{1}{k} \sum_{i=1}^{k} T_{\nu}(w^k) \leq \frac{1}{k} \sum_{i=1}^{k} \|\frac{1}{\nu} A(x^{i-1} - x^i)\|^2 \leq \frac{2}{\nu k} [p_{\nu}(w^0) - p_{\nu}^*],
\]
as required.
Lemma 13 Define a sequence $d^k = \frac{1}{\nu}(w^k - w^{k+1})$ based on the iterates generated by Algorithm 1. If Assumption 3 holds, then $p_\nu$ has a minimizer $w^*$, and

$$
\langle w^k - w^*, d^k \rangle \geq \frac{1}{2\nu} \|(I - P_A)(w^k - w^*)\|^2 + \frac{1}{\nu} \|\nu d^k\|^2 - \frac{1}{2\nu} \|\nu(I - P_A)d^k\|^2 + \frac{\alpha}{2} \|w^{k+1} - w^*\|^2.
$$

Proof [Lemma 13]

$$
p_\nu(w^{k+1}) = \frac{1}{2\nu} \|(I - P_A)w^{k+1}\|^2 + h(w^{k+1})
= \frac{1}{2\nu} \|(I - P_A)(w^k - \nu d^k)\|^2 + h(w^{k+1})
= \frac{1}{2\nu} \|(I - P_A)w^k\|^2 - \frac{1}{\nu} \langle \nu d^k, (I - P_A)w^k \rangle + \frac{1}{2\nu} \|\nu(1 - P_A)d^k\|^2 + h(w^{k+1})
$$

Decompose the first term above as follows:

$$
\frac{1}{2\nu} \|(I - P_A)w^k\|^2 = \frac{1}{2\nu} \|(I - P_A)(w^k - w^* + w^*)\|^2
= \frac{1}{2\nu} \|(I - P_A)(w^k - w^*)\|^2 + \frac{1}{\nu} \langle w^*, (I - P_A)(w^k - w^*) \rangle + \frac{1}{2\nu} \|(I - P_A)w^*\|^2
= -\frac{1}{2\nu} \|(I - P_A)(w^k - w^*)\|^2 + \frac{1}{\nu} \langle w^k - w^*, (I - P_A)w^* \rangle + \frac{1}{2\nu} \|(I - P_A)w^*\|^2
$$

Then we have,

$$
p_\nu(w^{k+1}) = -\frac{1}{2\nu} \|(I - P_A)(w^k - w^*)\|^2 + \frac{1}{\nu} \langle w^{k+1} - w^*, (I - P_A)w^k \rangle
+ \frac{1}{2\nu} \|(I - P_A)w^*\|^2 + \frac{1}{2\nu} \|\nu(1 - P_A)d^k\|^2 + h(w^{k+1})
$$

Since $h$ is convex and we know $d^k - \frac{1}{\nu}(w^k - P_Aw^k) \in \partial h(w^{k+1})$ we have,

$$
h(w^{k+1}) \leq h(w^*) + \frac{1}{\nu} \langle \nu d^k - (I - P_A)w^k, w^{k+1} - w^* \rangle - \frac{\alpha}{2} \|w^{k+1} - w^*\|^2
$$

Combining these results, we get

$$
p_\nu(w^{k+1}) \leq -\frac{1}{2\nu} \|(I - P_A)(w^k - w^*)\|^2 + \frac{1}{\nu} \langle w^{k+1} - w^*, \nu d^k \rangle
+ \frac{1}{2\nu} \|(I - P_A)w^*\|^2 + \frac{1}{2\nu} \|\nu(1 - P_A)d^k\|^2 + h(w^*) - \frac{\alpha}{2} \|w^{k+1} - w^*\|^2
$$

$$
0 \leq p_\nu(w^{k+1}) - p_\nu(w^*) \leq -\frac{1}{2\nu} \|(I - P_A)(w^k - w^*)\|^2 + \frac{1}{\nu} \langle w^k - w^*, \nu d^k \rangle
- \frac{1}{\nu} \|\nu d^k\|^2 + \frac{1}{2\nu} \|\nu(1 - P_A)d^k\|^2 - \frac{\alpha}{2} \|w^{k+1} - w^*\|^2
$$

which show the result:

$$
\langle w^k - w^*, d^k \rangle \geq \frac{1}{2\nu} \|(I - P_A)(w^k - w^*)\|^2 + \frac{1}{\nu} \|\nu d^k\|^2 - \frac{1}{2\nu} \|\nu(1 - P_A)d^k\|^2 + \frac{\alpha}{2} \|w^{k+1} - w^*\|^2.
$$
Proof [Theorem 7]
Using the same \( \{d^k\} \) as in Lemma 13,
\[
\|w^{k+1} - w^*\|^2 = \|w^k - \nu d^k - w^*\|^2
\]
\[
\|w^{k+1} - w^*\|^2 = \|w^k - w^*\|^2 - 2\left(\langle w^k - w^* , \nu d^k \rangle + \|\nu d^k\|^2\right)
\]
\[
(1 + \alpha \nu)\|w^{k+1} - w^*\|^2 \leq \|w^k - w^*\|^2 - \| (I - PA)(w^k - w^*) \|^2 - \|\nu d^k\|^2 + \|\nu(I - PA)d^k\|^2
\]
\[
(1 + \alpha \nu)\|w^{k+1} - w^*\|^2 \leq \|PA(w^k - w^*)\|^2 - \|PA(w^k - w^{k+1})\|^2
\]
\[
\|w^{k+1} - w^*\|^2 \leq \frac{1}{1 + \alpha \nu}\|w^k - w^*\|^2
\]

Lemma 14 If Assumption 4 holds, the iterates generated by the Algorithm 1 satisfy,
\[
\|PA(w^k - w^{k+1})\| \leq \|PA(w^k - w^*)\| \quad \forall k \in \mathbb{N}_+.
\]

Proof [Lemma 14]
Since \( w^{k+1} = \text{argmin}_w h(w) + \frac{1}{2\nu}\|w - PAw^k\|^2 \), we know,
\[
h(w^{k+1}) + \frac{1}{2\nu}\|w^{k+1} - PAw^k\|^2 \leq h(w^*) + \frac{1}{2\nu}\|w^* - PAw^k\|^2.
\]
By re-arranging terms, we get
\[
\|w^{k+1} - PAw^k\|^2 - \|(I - PA)w^{k+1}\|^2 - (\|w^* - PAw^k\|^2 - \|(I - PA)w^*\|^2) \leq 2\nu(g_\nu(w^*) - g_\nu(w^{k+1})) \leq 0.
\]
Since,
\[
\|(I - PA)w\|^2 + \|PA(w - w^k)\|^2 = \|w - PAw^k\|^2
\]
we have,
\[
\|w^{k+1} - PAw^k\|^2 - \|(I - PA)w^{k+1}\|^2 = \|PA(w^k - w^{k+1})\|^2
\]
\[
\|w^* - PAw^k\|^2 - \|(I - PA)w^*\|^2 = \|PA(w^k - w^*)\|^2
\]
Therefore,
\[
\|PA(w^k - w^{k+1})\| \leq \|PA(w^k - w^*)\| \quad \forall k \in \mathbb{N}_+.
\]

Lemma 15 Assume Assumption 4 holds, the iterates generated by the Algorithm 1 satisfy,
\[
\|w^{k+1} - w^*\|^2 \leq \|PA(w^k - w^*)\|^2 - \|\nu PA d^k\|^2.
\]
Moreover,
\[
\|w^{k+1} - w^*\| \leq \|PA(w^k - w^*)\|.
\]
Proof [Lemma 15] The proof uses the same technique as the proof of Theorem 7.

Proof [Theorem 9]
Since $w^{k+1} = \arg\min_w h(w) + \frac{1}{2\nu}\|w - PAw^k\|^2$, we know,

$$0 \in \partial h(w^{k+1}) + \frac{1}{\nu}(w^{k+1} - PAw^k)$$

$$\frac{1}{\nu}PA(w^k - w^{k+1}) \in \partial h(w^{k+1}) + \frac{1}{\nu}(w^{k+1} - PAw^{k+1})$$

$$\frac{1}{\nu}PA(w^k - w^{k+1}) \in \partial p_\nu(w^{k+1})$$

Because $p_\nu$ is convex and $w^*$ is a sharp minima,

$$\alpha\|w^{k+1} - w^*\| \leq p_\nu(w^{k+1}) - p_\nu(w^*) \leq \frac{1}{\nu} \langle PA(w^k - w^{k+1}), w^{k+1} - w^* \rangle$$

Expanding the right inequality we obtain

$$p_\nu(w^{k+1}) - p_\nu(w^*) \leq \frac{1}{\nu} \langle PA(w^k - w^{k+1}), w^{k+1} - w^* \rangle$$

$$\leq \frac{1}{\nu} \|PA(w^k - w^{k+1})\|^2 + \frac{1}{\nu} \langle PA(w^k - w^{k+1}), w^k - w^* \rangle$$

$$\leq \frac{1}{\nu} \|PA(w^k - w^{k+1})\| \|w^k - w^*\|$$

$$\leq \frac{1}{\nu} \|w^k - w^*\|^2$$

Therefore,

$$\|w^{k+1} - w^*\| \leq \frac{1}{\alpha\nu} \|w^k - w^*\|^2.$$ 

Combined with Lemma 15 we have, for all $k \geq K$,

$$\|w^{k+1} - w^*\| \leq \min \left\{ \|w^k - w^*\|, \frac{1}{\alpha\nu} \|w^k - w^*\|^2 \right\}$$

which gives the locally quadratic convergence rate.
**Proof** [Theorem 11]
We introduce a sequence \( \{x^k\} \) that satisfies,

\[
x^k = \arg\min_x \frac{1}{2\nu} \|Ax - w^k\| + g(x), \quad A^T(w^k - Ax^k) \in \nu \partial g(x^k), \quad \nu \nabla g_\nu(w^k) = w^k - Ax^k.
\]

Then we know the iterates of Algorithm 3 satisfy,

\[
\frac{1}{\nu} A(x^k - x^{k+1}) \in \nabla g_\nu(w^{k+1}) + \sum_{i=1}^m v_i^k \partial h_i(w^{k+1}),
\]

\[
\frac{1}{\alpha} (v^k - v^{k+1}) \in H(w^{k+1}) + \partial \delta(v^{k+1}|\triangle_r).
\]

By definition we know,

\[
p_\nu^k(w^{k+1}, v^k) = \sum_{i=1}^m v_i^k h_i(w_i^{k+1}) + g_\nu(w^{k+1})
\]

\[
= \sum_{i=1}^m v_i^k h_i(w_i^{k+1}) + \frac{1}{2\nu} \|Ax^{k+1} - w^{k+1}\|^2 + g(x^{k+1})
\]

\[
= \sum_{i=1}^m v_i^k h_i(w_i^{k+1}) + \frac{1}{2\nu} \|Ax^k - w^{k+1} + A(x^{k+1} - x^k)\|^2 + g(x^{k+1})
\]

\[
\leq \sum_{i=1}^m v_i^k h_i(w_i^{k}) + \frac{1}{2\nu} \|Ax^k - w^k\|^2
\]

\[
+ \frac{1}{\nu} \left\langle Ax^k - w^{k+1}, A(x^{k+1} - x^k) \right\rangle + \frac{1}{2\nu} \|A(x^{k+1} - x^k)\|^2 + g(x^{k+1})
\]

Since \( g \) is convex, we have,

\[
g(x^k) \geq g(x^{k+1}) + \frac{1}{\nu} \left\langle A^T(w^k - Ax^k), x^k - x^{k+1} \right\rangle
\]

\[
= g(x^{k+1}) + \frac{1}{\nu} \left\langle w^k - Ax^k, A(x^k - x^{k+1}) \right\rangle.
\]

Plug this inequality into the result above, we get

\[
p_\nu^k(w^{k+1}, v^k) \leq \sum_{i=1}^m v_i^k h_i(w_i^{k}) + \frac{1}{2\nu} \|Ax^k - w^k\|^2 + g(x^k) - \frac{1}{2\nu} \|A(x^k - x^{k+1})\|^2,
\]

\[
p_\nu^k(w^{k+1}, v^k) - p_\nu^k(w^k, v^k) \leq -\frac{1}{2\nu} \|A(x^k - x^{k+1})\|^2.
\]
An analogous calculation for $v$ gives us

$$p^t_v(w^{k+1}, v^{k+1}) - p^t_v(w^k, v^k)$$

$$= \langle H(w^{k+1}), v^{k+1} - v^k \rangle + \delta(v^{k+1}|\Delta_\tau) - \delta(v^k|\Delta_\tau)$$

$$= -\frac{1}{\alpha} \|v^{k+1} - v^k\|^2 - \left[\delta(v^k|\Delta_\tau) - \delta(v^{k+1}|\Delta_\tau) + \left(\partial\delta(v^{k+1}|\Delta_\tau), v^k - v^{k+1}\right)\right]$$

$$\leq -\frac{1}{\alpha} \|v^{k+1} - v^k\|^2$$

Therefore we can conclude that,

$$T^t_v(w^{k+1}, v^{k+1}) \leq \frac{1}{2\nu} \|A(x^k - x^{k+1})\|^2 + \frac{1}{\alpha} \|v^{k+1} - v^k\|^2$$

$$\leq p^t_v(w^k, v^k) - p^t_v(w^{k+1}, v^{k+1}) + p^t_v(w^{k+1}, v^k) - p^t_v(w^{k+1}, v^{k+1})$$

$$= p^t_v(w^k, v^k) - p^t_v(w^{k+1}, v^{k+1})$$

Adding up the telescoping series, we get the final result:

$$\frac{1}{k} \sum_{i=1}^k T^t_v(w^i, v^i) \leq \frac{1}{k} [p^t_v(w^0, v^0) - p^t_v(w^k, v^k)].$$

$\blacksquare$