A SMART Stochastic Algorithm for Nonconvex Optimization with Applications to Robust Machine Learning

Aleksandr Aravkin[*] SARAVKIN@UW.EDU
Department of Applied Mathematics
University of Washington
Seattle, WA 98195-4322, USA

Damek Davis[†] DSD95@CORNELL.EDU
School of Operations Research and Information Engineering
Cornell University
Ithaca, NY 14850, USA

Abstract

Machine learning theory typically assumes that training data is unbiased and not adversarially generated. When real training data deviates from these assumptions, trained models make erroneous predictions, sometimes with disastrous effects. Robust losses, such as the huber norm, were designed to mitigate the effects of such contaminated data, but they are limited to the regression context.

In this paper, we show how to transform any optimization problem that arises from fitting a machine learning model into one that (1) detects and removes contaminated data from the training set while (2) simultaneously fitting the trimmed model on the uncontaminated data that remains. To solve the resulting nonconvex optimization problem, we introduce a fast stochastic proximal-gradient algorithm that incorporates prior knowledge through nonsmooth regularization. For datasets of size n, our approach requires $O(n^{2/3}/\varepsilon)$ gradient evaluations to reach ε-accuracy and, when a certain error bound holds, the complexity improves to $O(\kappa n^{2/3}\log(1/\varepsilon))$. These rates are $n^{1/3}$ times better than those achieved by typical, full gradient methods.

Keywords: trimmed estimators, nonconvex optimization, SMART, SVRG, SAGA, variance reduction, machine learning

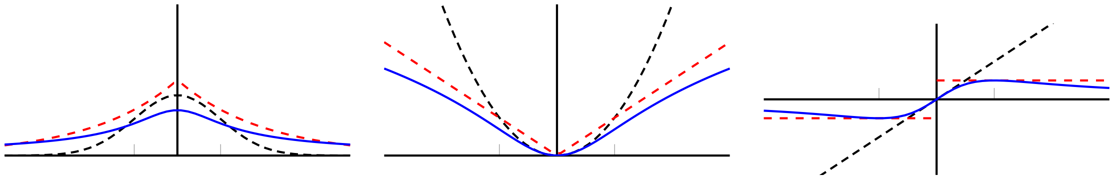
1. Introduction

Potential outliers in datasets can be identified in several ways. For low-dimensional models, scatter plots, box plots, and histograms can be used to visually identify points that deviate from modeling assumptions. For higher-dimensional data, several tests involving order statistics exist (so called L-estimators (Maronna et al., 2006)), such as the three-sigma rule for Gaussian data, or trimming strategies for disregarding points that are furthest away from the mean. After potential outliers are removed from a dataset, models are fit on the remaining data. After fitting the model, potential outliers are again identified and removed and another model is fit (Ruppert and Carroll, 1980). This process can repeat indefinitely, until no points are left in the dataset.

Identifying outliers using a fitted model can be problematic, since outliers affect the fit. Robust loss functions are often used to estimate model parameters from potentially contaminated data,

[*] This work was funded by the Washington Research Foundation Data Science Professorship.

[†] This material is based upon work supported by the National Science Foundation under Award No. 1502405.



(a) Densities (b) Penalties (c) Influence Functions

Figure 1: Gaussian (black, densely dashed), Huber(red dashed), and Student's t (blue solid).

without any *a priori* outlier removal or pre-processing. Examples include the ℓ_1, huber, and Student's t losses, all of which attempt to minimize the *influence* of observations that deviate from modeling assumptions (Huber, 2004; Lange et al., 1989), see Figure 1. After fitting a model using a robust loss, potential outliers can be identified by sorting the loss applied to individual observations. Observations with higher loss are considered more likely to be outliers.

Another approach, called trimmed estimation, couples explicit outlier identification and removal with model fitting. Given a set of n training examples, typical model fitting, i.e., M-estimation, solves

$$\underset{x}{\text{minimize}} \sum_{i=1}^{n} f_i(x),$$

where each f_i represents the loss associated with the ith training example. In contrast, trimmed M-estimators couple this already difficult, potentially nonconvex, optimization problem with explicit outlier removal

$$\underset{x}{\text{minimize}} \sum_{i=1}^{h} f_{i:n}(x), \tag{1}$$

where $f_{1:n}(x) \leq \cdots \leq f_{h:n}(x)$ are the first h order statistics of the objective values. If loss f_i is the log likelihood of the ith observed sample, then trimming attempts to jointly fit a probabilistic model while simultaneously eliminating the influence of all low likelihood observations.

Trimmed M-estimators were initially introduced by Rousseeuw (1985) in the context of least-squares regression. His original motivation was to develop linear regression estimators that have a high breakdown point (in this case 50%) and good statistical efficiency (in this case $n^{-1/2}$)[1]. These Least Trimmed Squares (LTS) estimators were proposed as a higher efficiency alternative to Least Median Squares (LMS) estimators (Rousseeuw, 1984), which replace the sum in (1) by a median. For a number of years, the difficulty of efficiently optimizing LTS problems limited their application. The problem is difficult because

even if all losses f_i are smooth and convex, (1) is, in general, nonsmooth and nonconvex.

Nevertheless, several approaches for finding LTS and other trimmed M-estimators have been developed. Rousseeuw and Van Driessen (2006) developed the FAST-LTS algorithm, which was able to find LTS estimators faster than existing algorithms for LMS estimations. Later, Mount

[1] Breakdown refers to the percentage of outlying points which can be added to a dataset before the resulting M-estimator can change in an unbounded way. Here, outliers can affect both the outcomes and training data (features).

et al. (2014) introduced an exact algorithm for computing LTS, which suffered from exponential complexity in higher dimensional problems. Generalizing the approach developed by Rousseeuw and Van Driessen (2006), Neykov and Müller (2003) developed the FAST-TLE method, which replaces the least squares terms in the LTS formulation with log-likelihoods of generalized linear models. In a different direction, Alfons et al. (2013) proposed a sparse variant of the Fast-LTS algorithm for L1-regularized LTS estimation. Further work by (Yang and Lozano, 2015; Yang et al., 2016) proposed algorithms for graphical lasso and regularized trimming of convex losses.

With the exception of Mount et al. (2014); Yang and Lozano (2015); Yang et al. (2016), each of the proposed algorithms above are variants of the alternating minimization algorithm. The algorithms by Yang and Lozano (2015); Yang et al. (2016) mixed alternating minimization and proximal-gradient steps. The algorithm of Mount et al. (2014) is combinatorial in nature, but has exponential complexity.

There are two drawbacks to trimming algorithms based on alternating minimization. First, they are greedy algorithms, which do not always work well for nonconvex problems; and second, they require, at every iteration, solving a large optimization problem typically involving more than 50% of the dataset.[2] The first drawback is well-known in the optimization community, while the second is motivation for introducing stochastic gradient approaches for trimming.

At first glance, the standard stochastic gradient (SG) method appears to be the natural algorithm for solving (1). However, (1) is nonsmooth and nonconvex, so there are, as of yet, no known convergence rate guarantees for SG applied to (1). In this paper we develop a variance-reduced stochastic gradient algorithm with convergence rate guarantees.

1.1 Contributions

Fully Nonconvex Problem Class. Our new algorithm extends the Stochastic Monotone Aggregated Root-Finding (SMART) algorithm (Davis, 2016b) to the nonsmooth, nonconvex trimming problem. To keep with tradition, we call this algorithm SMART. It is the first variance-reduced stochastic gradient algorithm for fully nonconvex optimization (our losses and our regularizers are nonconvex). It also applies to much more general problems than (1). We consider the following class:

$$\underset{w\in\mathbb{R}^n,\ x\in\mathcal{H}}{\text{minimize}} \left\{ \frac{1}{n}\sum_{i=1}^{n} w_i f_i(x) + r_1(w) + r_2(x) \right\}, \tag{2}$$

where each f_i is C^1 and r_1 and r_2 are lower semincontinuous (potentially nonconvex) functions. This more general problem class recovers (1): simply let $r_1 : \mathbb{R}^n \to [0, \infty]$ be the indicator function of the capped simplex

$$\varDelta_h := \left\{ (w_1, \ldots, w_n) \mid w \in [0,1], \sum_{i=1}^{n} w_i = h \right\},$$

and minimize jointly over w and x.

[2] For example, Alfons et al. (2013) requires solving a full LASSO problem at each iteration. And although the algorithm of (Yang et al., 2016) requires only one pass over the dataset at each iteration, this is still problematic for large datasets.

Better Dependence on Lipschitz Constants. It is possible to *apply* the proximal gradient algorithm to this problem[3] but its convergence is not guaranteed without taking incredibly small stepsizes. This restriction arises because the standard sufficient condition for guaranteeing the convergence of the proximal gradient method requires using a stepsize that is proportional to the inverse of the Lipschitz constant of the gradient of the smooth function $G(w, x) = w_i f_i(x)$, which are not globally Lipschitz: $\nabla G(w, x) = (f_i(x), w_i \nabla f_i(x))$. Even for least squares problems, the local Lipschitz constant of $\nabla G(w, x)$ grows with $\|x\|$ and $\|w\|$. This issue likewise prevents our using the ProxSAGA and ProxSVRG (Reddi et al., 2016).

Convergence Rates that Scale with $n^{2/3}$. A good alternative to the proximal-gradient method is called the Proximal Alternating Linearized Minimization (PALM) method (Bolte et al., 2014) (see Section 2), which allows for stepsizes that only scale inversely with $\|w\|$ and the Lipschitz constants of ∇f_i. The convergence rate of this algorithm was analyzed in the fully nonconvex case in (Davis, 2016a, Theorem 5.4), where it was shown that an ε-stationary point (see Section 3.1) could be found within $O(1/\varepsilon)$ iterations. Thus, in total PALM finds ε-stationary points using $O(n/\varepsilon)$ gradients.

SMART scales better than PALM and other competing methods by a factor of $n^{1/3}$. In particular, without any regularity assumptions

SMART finds an ε-stationary point in in $O(n + n^{2/3}/\varepsilon)$ gradient evaluations

(see Corollaries 2 and 3). This matches the complexity of ProxSAGA and ProxSVRG (Reddi et al., 2016), which only apply to the special case of problem (2) considered in Section 2.2.

When a certain error bound holds (see (5)),

SMART finds an ε-stationary point in $O\left(n + \kappa n^{2/3} \log(1/\varepsilon)\right)$ gradient evaluations.

where κ is akin to a condition number of (2) (see Corollaries 5 and 6). In contrast, ProxSAGA and ProxSVRG (Reddi et al., 2016), which only apply to the special case of problem (2) considered in Section 2.2, both require $O((n + \kappa n^{2/3}) \log(1/\varepsilon))$ gradient evaluations to reach accuracy ε.

Organization. We present algorithms related to SMART in Sections 2.2 and 2.3. We also present several theoretical guarantees for SMART in Section 3. In Section 4, we illustrate three applications of trimming to robust estimation. We present robust digit recognition for the **mnist** dataset, introduce trimmed Principal Component Analysis to determine the quality of judges in the **USJudges** dataset, and apply SMART to find a homography between two images using interest point matching. Proofs of the main theorems are presented in the appendices.

1.2 Notation

In Problem (2) the variable x is an element of a finite dimensional Euclidean space $\mathcal{H}$; each function $f_i : \mathcal{H} \to \mathbb{R}$ is C^1, each gradient ∇f_i is L-Lipschitz continuous; both functions $r_1 : \mathbb{R}^n \to (-\infty, \infty]$ and $r_2 : \mathcal{H} \to (-\infty, \infty]$ are proper and lower-semicontinuous. We assume that the point-to-set proximal mapping $\mathbf{prox}_{\gamma r_j} : \mathcal{H} \to 2^{\mathcal{H}}$

$$x \mapsto \operatorname*{argmin}_{x' \in \mathcal{H}} \left\{ r_j(x') + \frac{1}{2\eta}\|x' - x\|^2 \right\}$$

[3] For example, the pioneering work of Attouch et al. (2013) proved that the proximal gradient algorithm converges under extremely general conditions.

is always nonempty for every η small enough, say for $\eta < \delta_{r_1}$ if $j = 1$ and for $\eta < \delta_{r_2}$ if $j = 2$.

We work with an underlying probability space denoted by $(\Omega, \mathcal{F}, P)$, and we assume that the space $\mathcal{H}$ is equipped with Borel σ-algebra $\mathcal{B}$. An $\mathcal{H}$-valued random variable is a measurable map $X : (\Omega, \mathcal{F}) \to (\mathcal{H}, \mathcal{B})$. We always let $\sigma(X) \subseteq \mathcal{F}$ denote the sub σ-algebra generated by a random variable X. We use the shorthand $\mathrm{a.s.}$ to denote almost sure convergence of a sequence of random variables. By our assumptions on r_1 and r_2, for $j \in \{1, 2\}$ there exists measurable mappings $\zeta_j : \mathcal{H}_j \times (0, \delta_{r_j}) \to \mathcal{H}_j$ such that $\zeta_j(x, \gamma) \in \mathbf{prox}_{\gamma r_j}(x)$ for all $(x, \gamma) \in \mathcal{H}_j \times (0, \delta_{r_j})$, where $\mathcal{H}_1 = \mathbb{R}^n$ and $\mathcal{H}_2 = \mathcal{H}$ (Rockafellar and Wets, 1998). For the rest of the paper, we let $x^+ = \mathbf{prox}_{\gamma r_j}(x)$ mean that $x^+ = \zeta_j(x, \gamma)$.

We use the notation

$$F(w, x) = \frac{1}{n} \sum_{i=1}^{n} w_i f_i(x) + r_1(w) + r_2(x)$$

throughout the paper and assume that $(w^*, x^*) \in \operatorname{argmin}_{x \in \mathcal{H}, w \in \mathbb{R}^n} F(w, z)$ exists.

We assume that $\mathrm{dom}(r_1)$ is bounded: there exists $B_i > 0$ such that for all $w \in \mathrm{dom}(r_1)$, we have $|w_i| \le B_i$.

2. Algorithm

To find a stationary point of (2), our algorithm iteratively updates a state vector $(w^k, x^k) \in \mathbb{R}^n \times \mathcal{H}$. The algorithm is designed so that (w^k, x^k) will not only be close to a stationary point after just a few iterations, but so that the average computational complexity of obtaining (w^{k+1}, x^{k+1}) from (w^k, x^k) will be small. These competing objectives can both be achieved simultaneously by combining ideas from the *Proximal Alternating Linearized Minimization* (PALM) method (Bolte et al., 2014), which obtains (w^{k+1}, x^{k+1}) from (w^k, x^k) via

$$\begin{aligned} w^{k+1} &:= \mathbf{prox}_{\tau r_1}\left(w^k - \frac{\tau}{n}(f_1(x^k), \ldots, f_n(x^k))\right); \\ x^{k+1} &:= \mathbf{prox}_{\gamma r_2}\left(x^k - \frac{\gamma}{n}\sum_{i=1}^{n} w_i^{k+1} \nabla f_i(x^k)\right), \end{aligned}$$

and the *partially stochastic proximal-gradient* (PSPG) method, which obtains (w^{k+1}, x^{k+1}) from (w^k, x^k) via

$$\begin{aligned} w^{k+1} &:= \mathbf{prox}_{\tau r_1}\left(w^k - \frac{\tau}{n}(f_1(x^k), \ldots, f_n(x^k))\right); \\ x^{k+1} &:= \mathbf{prox}_{(\gamma/\eta) r_2}\left(x^k - \frac{\gamma}{n} w_{i_k}^k \nabla f_{i_k}(x^k)\right), \end{aligned}$$

where $i_k \in \{1, \ldots, n\}$ is randomly sampled and $\gamma \to 0$ as $k \to \infty$.

PALM takes few iterations to obtain near stationary (w^k, x^k) (ε accuracy obtained after $O(1/\varepsilon)$ iterations), but for each k it computes the full gradient $n^{-1} \sum_{i=1}^{n} w_i^k \nabla f_i(x^k)$, which can be costly. On the other hand, PSPG takes many iterations to obtain near stationary (w^k, x^k), but for each k it only computes a single gradient $w_{i_k}^k \nabla f_{i_k}(x^k)$, which can be done quickly. But for nonconvex problems, **there is no known rate of convergence for PSPG** (unless minibatches of stochastic gradients of increasing size are used (Ghadimi et al., 2016; Davis et al., 2016)). Even in the relatively simple case where f_i, r_1, and r_2 are convex, there is still a nonconvex coupling between w_i and f_i and, hence, no known rate of convergence for PSPG.

By reducing the variance of the stochastic gradient estimator $w_{i_k}\nabla f_{i_k}$, we create a fast algorithm, which we call SMART, that combines the PALM and PSPG updates and obtains an ε accuracy solution after $O(1/\varepsilon)$ steps. As in PSPG, SMART typically evaluates a single gradient ∇f_{i_k} (or a small batch) at one or two points per iteration. But unlike PSPG, SMART on average only evaluates all the function values $(f_1(x^k), \ldots, f_n(x^k))$ once per every t iterations, where t is user defined.

2.1 Implementation and Features

Algorithm 1 SMART for (2)

```
Choose $\gamma < \delta_{r_1}$;  $\tau < \delta_{r_2}$;  $(w^0, x^0) \in \mathrm{dom}(r_1) \times \mathrm{dom}(r_2)$;  $y_i^0 = \nabla f_i(x^0)^T w_i^0$.
for $k = 0, 1, \ldots$ do
    Sample $I_k \subseteq \{1, \ldots, n\}$; $j_k \in \{1, 2\}$; $D_k \subseteq \{1, \ldots, n\}$;
    if $j_k = 1$ then
        $w^{k+1} \leftarrow \mathbf{prox}_{\tau r_1}\left(w^k - \frac{\tau}{n}(f_1(x^k), \ldots, f_n(x^k))\right)$;
        $x^{k+1} \leftarrow x^k$;
        for $i = 1, \ldots, n$ do
            $y_i^{k+1} \leftarrow w_i^{k+1}\nabla f_i(x^k)$;
        end for
    else
        $w^{k+1} \leftarrow w^k$;
        $x^{k+1} \leftarrow \mathbf{prox}_{\gamma r_2}\left(x^k - \gamma\left(\frac{1}{b}\sum_{i \in I_k}(w_i^k \nabla f_i(x^k) - y_i^k) + \frac{1}{n}\sum_{i=1}^n y_i^k\right)\right)$;
        for $i \in D_k$ do
            $y_i^{k+1} \leftarrow w_i^k \nabla f_i(x^k)$;
        end for
    end if
end for
```

Incremental Gradients and Minibatches. Rather than evaluating a full gradient $\nabla f = n^{-1}\sum_{i=1}^n \nabla f_i$ at each iteration, we instead sample b elements uniformly at random with replacement and denote this collection by $I_k \subseteq \{1, \ldots, n\}$; then we only evaluate ∇f_i for $i \in I_k$. We assume that $\{I_k\}_{k\in\mathbb{N}}$ is IID.

Block Coordinates Updates. At every iteration we sample a coordinate $j_k \subseteq \{1, 2\}$ that indicates whether w^k is modified ($j_k = 1$) or whether x^k is modified ($j_k = 2$) to obtain (w^{k+1}, x^{k+1}). We assume that $\{j_k\}_{k\in\mathbb{N}}$ is IID and the variables I_k and j_k are independent. We let

$$q := P(j_k = 1) > 0, \qquad\qquad q' = P(j_k = 2) > 0.$$

Dual Variables and Dual Updates. For each index $i \in \{1, \ldots, n\}$, we maintain a sequence of *dual variables*, denoted by $y_i^k \in \mathcal{H}$. The dual variables are always parametrically defined: $y_i^k = \phi_{i1}^k \nabla f_i(\phi_{i2}^k)$ for old iterates $(\phi_{i1}^k, \phi_{i2}^k) \in \{(w_i^l, x_i^l)\}_{l<k}$. The sum $n^{-1}\sum_{i=1}^n y_i^k$ approximates

the gradient $n^{-1} \sum_{i=1}^{n} w_i^k \nabla f_i(x^k)$ and is used in the following stochastic estimator of the sum, which has smaller variance than the SG estimator $w_i^k \nabla f_i(x^k)$:

$$\frac{1}{b} \sum_{i \in I_k} (w_i^k \nabla f_i(x^k) - y_i^k) + \frac{1}{n} \sum_{i=1}^{n} y_i^k. \tag{3}$$

The dual variables need not be recomputed at every iteration, so ϕ_i^k can be quite a stale estimate of x^k. We introduce the set-valued random variable

$$D_k \subseteq \{1, \ldots, n\}$$

and the probability

$$\rho_i := P(i \in D_k)$$

which control whether the ith dual variable is updated at iteration k:

$$y_i^{k+1} = \begin{cases} w_i^k \nabla f_i(x^k) & \text{if } i \in D_k; \\ y_i^k & \text{otherwise.} \end{cases}$$

We assume that $\{D_k\}_{k \in \mathbb{N}}$ is IID and that D_k is independent from j_k, but we do not assume that D_k is independent from I_k.

2.2 Connection to ProxSAGA and ProxSVRG.

Our main goal is to use the regularizer r_1 to trim statistical models, but we can trivially turn off trimming by choosing

$$r_1(w) = \begin{cases} 0 & \text{if } w_i \neq 1 \text{ for } i = 1, \ldots, n; \\ \infty & \text{otherwise,} \end{cases}$$

which forces all weights w_i to be 1.

In this case, we recover and extend the ProxSAGA algorithm, introduced by Defazio et al. (2014) and recently analyzed for nonconvex problems by Reddi et al. (2016), by letting D_k be a set consisting of b elements of $\{1, \ldots, n\}$, sampled uniformly at random with replacement, and by letting $q = 0$. In terms of implementation, we never perform a w or a full gradient update, but at every iteration we update the dual variable y_i^k for $i \in D_k$. **Our work extends the work by Reddi et al. (2016) by allowing nonconvex regualizers** r_2, whereas Reddi et al. (2016) require r_2 to be convex.

We also recover a variant of ProxSVRG, introduced by Xiao and Zhang (2014) and recently analyzed for nonconvex problems analyzed by Reddi et al. (2016), by setting $D_k = \emptyset$ and $q = 1/t$, where $t > 1$ is the average number of iterations we wish to perform before recomputing a full gradient. Although it appears that the w step requires a computation of the function values $(f_1(x^k), \ldots, f_n(x^k))$, it does not because $w_i^k \equiv 1$. As in the ProxSAGA case, our work extends Reddi et al. (2016) by allowing nonconvex regularizers r_2.

Algorithm	**GradEvals**	**FunEvals**	$\mathbf{prox}_{\tau r_1}$**Evals**	$\mathbf{prox}_{\gamma r_2}$**Evals**
SMART(SAGA)	$O(n + \frac{n^{2/3}}{\varepsilon})$	$O(\frac{1}{\varepsilon})$	$O(\frac{1}{n\varepsilon})$	$O(\frac{1}{\varepsilon})$
SMART(SAGA+(5))	$O(n + \kappa n^{2/3} \log(\frac{1}{\varepsilon}))$	$O(\kappa \log(\frac{1}{\varepsilon}))$	$O(\kappa \log(\frac{1}{\varepsilon}))$	$O(\frac{\kappa}{n} \log(\frac{1}{\varepsilon}))$
SMART(SVRG)	$O(n + \frac{n^{2/3}}{\varepsilon})$	$O(\frac{n^{2/3}}{\varepsilon})$	$O(\frac{1}{n^{1/3}\varepsilon})$	$O(\frac{1}{\varepsilon})$
SMART(SVRG+(5))	$O(n + \kappa n^{2/3} \log(\frac{1}{\varepsilon}))$	$O(\kappa n^{2/3} \log(\frac{1}{\varepsilon}))$	$O(\frac{\kappa}{n^{1/3}} \log(\frac{1}{\varepsilon}))$	$O(\kappa \log(\frac{1}{\varepsilon}))$
PALM	$O(\frac{n}{\varepsilon})$	$O(\frac{1}{\varepsilon})$	$O(\frac{1}{\varepsilon})$	$O(\frac{1}{\varepsilon})$
PALM(+(5))	$O(\kappa n \log(\frac{1}{\varepsilon}))$	$O(\kappa n \log(\frac{1}{\varepsilon}))$	$O(\kappa \log(\frac{1}{\varepsilon}))$	$O(\kappa \log(\frac{1}{\varepsilon}))$

Table 1: Convergence rates of SMART and PALM in terms of number of operations needed to achieve accuracy ε. The constant κ is defined in Section 3.2. The rates for SMART are proved in Corollaries 2, 3, 5, and 6. The rates for PALM can be determined (with some effort) from the proofs in (Davis, 2016a). Alternatively, the rates for PALM may be derived from Theorems 1 and 4 by using the randomized variant of PALM discussed in Section 2.3.

2.3 Connection to Partial Minimization and Randomized Coordinate Descent

With appropriate choices of the random variables j_k, I_k, and D_k, we recover randomized variants of PALM (Bolte et al., 2014) and the full gradient method of Aravkin et al. (2016). The key is to choose $I_k = D_k \equiv \{1, \ldots, n\}$, so that all dual variables are constantly updated, and $q := P(j_k = 1) = 1/2$. Then, our stochastic estimator (3) is equal to the full gradient:

$$\frac{1}{n} \sum_{i=1}^{n} w_i^k \nabla_i f_i(x^k).$$

For fixed τ, we get a randomized variant of the algorithm of Bolte et al. (2014). For $\tau \to \infty$, we get a method similar to that of Aravkin et al. (2016), except that we allow nonconvex regularizers. When r_2 is convex, $\mathbf{prox}_{\tau r_2}(w)$ converges to an element of $\operatorname{argmin}\{r_2(w)\}$ (Bauschke and Combettes, 2011, Theorem 23.44); in the general case r_2 need only be prox bounded, so $\mathbf{prox}_{\tau r_2}$ may not even be defined for large τ.

3. Convergence Theory

Our convergence rates are organized in Table 1. We separate our sublinear and linear convergence rate results into Section 3.1 and 3.2, respectively.

3.1 Sublinear Rates

ε-Stationary Points. For all $k \in \mathbb{N}$, we define vectors $\overline{x}^{k+1} \in \mathcal{H}$ and $\overline{w}^{k+1} \in \mathbb{R}^n$ by:

$$\overline{w}^{k+1} := \mathbf{prox}_{\tau r_1}\left(w^k - \frac{\tau}{n}(f_1(x^k), \ldots, f_n(x^k))\right)$$
$$\overline{x}^{k+1} := \mathbf{prox}_{(\gamma/\eta) r_2}\left(x^k - \frac{\gamma}{\eta n}\sum_{i=1}^{n} w_i^k \nabla f_i(x^k)\right).$$

SMART never actually computes $\overline{x}^{k+1}$; it is only used in the analysis of the algorithm. Its existence shows that a nearby, nearly stationary point can be obtained with n gradient evaluations. For our analysis, it is crucial that η be a constant greater than 1, i.e., we must shorten the steplength in order to measure stationarity.

We measure convergence of (w^k, x^k) by bounding the normalized step sizes

$$\begin{aligned} \frac{1}{\tau}\left(w^k - \overline{w}^{k+1}\right) &\in \frac{1}{n}(f_1(x^k), \ldots, f_n(x^k)) + \partial_L r_1(\overline{w}^{k+1}); \\ \frac{\eta}{\gamma}\left(x^k - \overline{x}^{k+1}\right) &\in \frac{1}{n}\sum_{i=1}^n w_i^k \nabla f_i(x^k) + \partial_L r_2(\overline{x}^{k+1}), \end{aligned}$$

where $\partial_L r_j$ denotes the *limiting subdifferential* of r_j (Rockafellar and Wets, 1998, Definition 8.3). It is common to compute bounds on the square of these step lengths, although it is perhaps misleading to do so. To make it easy to compare our results with the current literature, we also bound the squared steplengths in Part 3 of Theorem 1.

Using the Lipschitz continuity of ∇f_i and the local Lipschitz continuity of f_i, these bounds easily translate bounds on

$$\operatorname{dist}\left(0, \partial_L F(\overline{w}^{k+1}, \overline{x}^{k+1})\right).$$

We omit this straightforward derivation.

Independence of Algorithm History and Sampling The SMART algorithm generates a sequence of random variables $\{(w^k, x^k)\}_{k\in\mathbb{N}}$. Throughout the algorithm, we make the standard assumption that

Assumption 1 *The σ-algebra generated by the history of SMART, denoted by*

$$\mathcal{F}_k = \sigma((w^0, x^0), \ldots, (w^k, x^k)),$$

is independent of the σ-algebra $\mathcal{I}_k = \sigma((I_k, j_k, D_k))$.

SMART converges, provided we choose γ properly. In measuring convergence, we introduce a particular $\eta > 0$ (which depends on a user defined constant $\epsilon_0 \in (0, 1)$):

$$\eta = 2\left(1 + 2\gamma\left[\sqrt{\frac{1}{n}\sum_{i=1}^n \frac{q'(1+\epsilon_0)(B_i L)^2}{2b\left(1 - \sqrt{q'(1-\rho_i)}\right)^2}} + \frac{L}{n}\sum_{i=1}^n B_i\right]\right). \tag{4}$$

This constant is key for showing that Algorithm 1 converges with nonconvex regularizers r_1 and r_2. We place the proof in Appendix A.

Theorem 1 (SMART Converges) *Suppose $\{(w^k, x^k)\}_{k\in\mathbb{N}}$ is generated by Algorithm 1 and that Assumption 1 holds. Let $\epsilon_0 \in (0, 1)$ and let η be defined as in* (4). *Then, if*

$$\gamma \le \frac{1}{4L\sqrt{\frac{1}{n}\sum_{i=1}^n \frac{q'(1+\epsilon_0)B_i^2}{2b\left(1-\sqrt{q'(1-\rho_i)}\right)^2}} + \frac{L}{n}\sum_{i=1}^n B_i},$$

the following hold:

1. ***Objective Decrease.*** *The limit* $\lim_{k\to\infty} F(w^k, x^k)$ *exists almost surely and for all* $k \in \mathbb{N}$*, we have*
$$\mathbb{E}\left[F(w^{k+1}, x^{k+1}) \mid \mathcal{F}_k\right] \leq F(w^0, x^0) \\ - \sum_{t=0}^{k} \left[\frac{q'\gamma}{2\eta}\left\|\frac{\eta}{\gamma}\left(x^t - \overline{x}^{t+1}\right)\right\|^2 + \frac{q\tau}{2}\left\|\frac{1}{\tau}\left(w^t - \overline{w}^{t+1}\right)\right\|^2\right].$$
2. ***Limit Points are Stationary.*** *Suppose that the sequence* $\{(w^k, x^k)\}_{k\in\mathbb{N}}$ *is almost surely bounded. Then* $F(\overline{w}^{k+1}, \overline{x}^{k+1})$ *converges almost surely to a random variable. Moreover, there exists a subset* $\widetilde{\Omega} \subseteq \Omega$ *such that* $P(\widetilde{\Omega}) = 1$ *and for all* $\omega \in \widetilde{\Omega}$*, every limit point of* $\{(\overline{w}^k(\omega), \overline{x}^k(\omega))\}_{k\in\mathbb{N}}$ *converges to a stationary point of* F*.*
3. ***Convergence Rate.*** *Fix* $T \in \mathbb{N}$*. Sample* t_0 *uniformly at random from* $t_0 \in \{0, \ldots, T\}$*. Then*
$$\frac{q'\gamma}{2\eta}\mathbb{E}\left[\left\|\frac{\eta}{\gamma}\left(x^{t_0} - \overline{x}^{t_0+1}\right)\right\|^2\right] + \frac{q\tau}{2}\mathbb{E}\left[\left\|\frac{1}{\tau}\left(w^{t_0} - \overline{w}^{t_0+1}\right)\right\|^2\right] \leq \frac{F(w^0, x^0) - F(w^*, x^*)}{T}.$$

With proper choices of b, **we actually achieve an ε-accuracy solution with fewer gradient and function evaluations than the proximal gradient method or PALM (Bolte et al., 2014)**, which require $O(n/\varepsilon)$ gradient evaluations and $O(n/\varepsilon)$ function evaluations.

The first corollary, whose proof is given Appendix A.1, applies to a variant of the ProxSAGA algorithm:

Corollary 2 (Convergence Rate of SAGA Variant of SMART) *Suppose that* $D_k \equiv I_k$*,* $q' = 1 - 1/n$

$$\gamma = \frac{1}{4L\sqrt{\frac{1}{n}\sum_{i=1}^n \frac{(1-1/n)(1+\epsilon_0)B_i^2}{2b\left(1-\sqrt{(1-1/n)^{b+1}}\right)^2} + \frac{L}{n}\sum_{i=1}^n B_i}}, \qquad \text{and} \qquad \tau = \frac{(n-1)\gamma}{\eta}.$$

Then SMART achieves an $\varepsilon > 0$ *accurate solution with, on average,* $O(n + n/(b^{3/2}\varepsilon) + n/(b^{1/2}\varepsilon))$ *gradient evaluations,* $O(n/(b^{3/2}\varepsilon))$ *evaluations of* $\mathbf{prox}_{\gamma r_2}$*,* $O(n/(b^{3/2}\varepsilon))$ *function evaluations, and* $O(1/(b^{3/2}\varepsilon))$ *evaluations of* $\mathbf{prox}_{\tau r_1}$*. In particular, when* $b = n^{2/3}$*, SMART achieves an* $\varepsilon > 0$ *accurate solution with, on average,* $O(n + n^{2/3}/\varepsilon)$ *gradient evaluations,* $O(1/\varepsilon)$ $\mathbf{prox}_{\gamma r_2}$ *evaluations,* $O(1/\varepsilon)$ *function evaluations, and* $O(1/(n\varepsilon))$ $\mathbf{prox}_{\gamma r_1}$ *evaluations.*

The second corollary, whose proof is given Appendix A.2, applies to a variant of the ProxSVRG algorithm:

Corollary 3 (Convergence Rate of SVRG Variant of SMART) *Suppose that* $D_k \equiv \emptyset$*,* $q' = (1 - 1/n)^b$*,*

$$\gamma = \frac{1}{4L\sqrt{\frac{1}{n}\sum_{i=1}^n \frac{(1-1/n)^b(1+\epsilon_0)B_i^2}{2b\left(1-\sqrt{(1-1/n)^b}\right)^2} + \frac{L}{n}\sum_{i=1}^n B_i}}, \quad \text{and} \quad \tau = \frac{(1-(1-1/n)^b)(1-1/n)^b\gamma}{\eta}.$$

Then SMART achieves an $\varepsilon > 0$ accurate solution with, on average, $O(n+n/(b^{1/2}\varepsilon))$ gradient evaluations, $O(n/(b^{3/2}\varepsilon))$ evaluations of $\mathbf{prox}_{\gamma r_2}$, $O(n/(b^{1/2}\varepsilon))$ function evaluations, and $O(1/(b^{1/2}\varepsilon))$ evaluations of $\mathbf{prox}_{\tau r_2}$. In particular, when $b = n^{2/3}$, SMART achieves an $\varepsilon > 0$ accurate solution with, on average, $O(n + n^{2/3}/\varepsilon)$ gradient evaluations, $O(1/\varepsilon)$ $\mathbf{prox}_{\gamma r_2}$ evaluations, $O(n^{2/3}/\varepsilon)$ function evaluations, and $O(1/(n^{1/3}\varepsilon))$ $\mathbf{prox}_{\tau r_1}$ evaluations.

3.2 Linear Rates

Assuming that an error bound holds for all points $(w, x) \in \operatorname{dom}(r_1) \times \operatorname{dom}(r_2)$, a potentially bounded set, we can prove stronger convergence rates.

The Global Error Bound. In our convergence analysis, we use a modified globalization of the error bound found in Drusvyatskiy and Lewis (2016). We assume that there exists $(w^*, x^*) \in \operatorname{dom}(r_1) \times \operatorname{dom}(r_2)$ such that for all $(w, x) \in \operatorname{dom}(r_1) \times \operatorname{dom}(r_2)$, we have

$$\left\|\frac{\eta}{\gamma}\left(x - \mathbf{prox}_{(\gamma/\eta)r_1}\left(x - \frac{\gamma}{\eta}\sum_{i=1}^{n} w_i \nabla f_i(x)\right)\right)\right\|^2 + \left\|\frac{1}{\tau}\left(w - \mathbf{prox}_{\tau r_2}(x - \tau(f_1(x), \ldots, f_n(x)))\right)\right\|^2 \\ \geq \mu\left[F(w,x) - F(w^*, x^*)\right] \qquad (5)$$

Drusvyatskiy and Lewis (2016) use a localized version of (5) to prove linear convergence of a proximal algorithm for minimizing convex composite objectives. Our error bound differs from their error bound in two ways: (1) their bound is only assumed to hold locally around critical points of F; and (2) their right hand side is $\mu\left[\|x - x^*\|^2 + \|w - w^*\|^2\right]$, rather than $\mu\left[F(w,x) - F(w^*, x^*)\right]$. We use this simplified error bound to keep the presentation short, but in future work, we may study the behavior of SMART assuming the localized bound in Drusvyatskiy and Lewis (2016).[4]

As in the sublinear case, we define a constant η (which depends on a user defined constant $\epsilon_0 \in (0, 1)$):

$$\eta = 2\left(1 + 2\gamma\left[\sqrt{\frac{1}{n}\sum_{i=1}^{n}\frac{q'(1+\epsilon_0)(B_iL)^2}{2b\sqrt{q'(1-\rho_i)}\left(1 - (q'(1-\rho_i))^{1/4}\right)^2}} + \frac{L}{n}\sum_{i=1}^{n} B_i\right]\right). \qquad (6)$$

The ratio γ/η controls the linear convergence rate of SMART.

Theorem 4 (Convergence Rate of SMART Assuming a Global Error Bound) *Assume the notation of Theorem 1. Let $\epsilon_0 \in (0, 1)$, let η be defined as in (6), and let*

$$\gamma = \frac{1}{4L\sqrt{\frac{1}{n}\sum_{i=1}^{n}\frac{q'(1+\epsilon_0)B_i^2}{2b\sqrt{q'(1-\rho_i)}\left(1-(q'(1-\rho_i))^{1/4}\right)^2}} + \frac{L}{n}\sum_{i=1}^{n} B_i}.$$

[4] Equation (5) is also quite similar to the *Kurdyka-Łojasiewicz* (KL) inequality with exponent $\frac{1}{2}$ (Bolte et al., 2007a,b), which replaces the left hand side of (5) by $\operatorname{dist}(0, \partial_L F(w,x))^2$. Its straightforward to prove linear convergence of SMART under this globalized KL error bound, but we omit it to keep the presentation short.

Define

$$\delta := \max_i \left\{ 1 - \mu \min\left\{ \frac{q'\gamma}{2\eta}, \frac{q\tau}{2} \right\}, \sqrt{q'(1-\rho_i)} \right\} \in (0,1).$$

Then provided that the error bound (5) *holds, then we have*

$$(\forall k \in \mathbb{N}) \qquad \mathbb{E}\left[F(w^k, x^k) - F(w^*, x^*) \right] \leq \delta^k \left[F(w^0, x^0) - F(w^*, x^*) \right].$$

By assuming an error bound similar to (5) and employing a restart strategy, Reddi et al. (2016) developed a linearly converging variant of ProxSAGA and ProxSVRG . In this strategy, the authors ran ProxSAGA or ProxSVRG for $\lceil 30\kappa \rceil$ iterations, where κ is akin to the inverse condition number

$$\kappa = \frac{L}{\mu},$$

before restarting the algorithm. Every time that ProxSAGA or ProxSVRG is restarted, a full gradient must be computed. **In contrast, SMART never needs to be restarted: it simply adapts to the regularity of the problem at hand.**

Frequent restarts of ProxSAGA and ProxSVRG lead to worse complexity. In both of the corollaries below, we show SMART needs $O(n + n^{2/3}\kappa \log(1/\varepsilon))$ gradients to reach accuracy ε. In contrast, ProxSAGA/SVRG need $O((n + n^{2/3}\kappa)\log(1/\varepsilon))$ gradients to reach accuracy ε.

The first corollary, whose proof is given Appendix B.1, applies to a variant of the ProxSAGA algorithm:

Corollary 5 (Linear Convergence Rate of SAGA Variant of SMART) *Suppose that* $D_k \equiv I_k$, $q' = 1 - 1/n$, *that* γ *is chosen as in Theorem 4, and* $\tau = \frac{(n-1)\gamma}{\eta}$. *Then SMART achieves an* $\varepsilon > 0$ *accurate solution with, on average,* $O(n + \kappa(n/b^{3/2} + n/b^{1/2})\log(1/\varepsilon))$ *gradient evaluations,* $O((\kappa n/b^{3/2})\log(1/\varepsilon))$ *evaluations of* $\mathbf{prox}_{\gamma r_2}$, $O((\kappa n/b^{3/2})\log(1/\varepsilon))$ *function evaluations, and* $O((\kappa/b^{3/2})\log(1/\varepsilon))$ *evaluations of* $\mathbf{prox}_{\tau r_1}$. *In particular, when* $b = n^{2/3}$, *SMART achieves an* $\varepsilon > 0$ *accurate solution with, on average,* $O(n + \kappa n^{2/3}\log(1/\varepsilon))$ *gradient evaluations,* $O(\kappa \log(1/\varepsilon))$ $\mathbf{prox}_{\gamma r_2}$ *evaluations,* $O(\kappa\log(1/\varepsilon))$ *function evaluations, and* $O((\kappa/n)\log(1/\varepsilon))$ $\mathbf{prox}_{\tau r_1}$ *evaluations.*

The second corollary, whose proof is a straightforward modification of the proof of Corollaries 5 and 3, applies to a variant of the ProxSVRG algorithm:

Corollary 6 (Linear Convergence Rate of SVRG Variant of SMART) *Suppose that* $D_k \equiv \emptyset$, $q' = (1 - 1/n)^b$, *that* γ *is chosen as in Theorem 4, and that* $\tau = \frac{(1-(1-1/n)^b)(1-1/n)^b\gamma}{\eta}$. *Then SMART achieves an* $\varepsilon > 0$ *accurate solution with, on average,* $O(n + (\kappa n/b^{1/2})\log(1/\varepsilon))$ *gradient evaluations,* $O((\kappa n/b^{3/2})\log(1/\varepsilon))$ *evaluations of* $\mathbf{prox}_{\gamma r_2}$, $O((n/b^{1/2})\log(1/\varepsilon))$ *function evaluations, and* $O((\kappa/b^{1/2})\log(1/\varepsilon))$ *evaluations of* $\mathbf{prox}_{\tau r_2}$. *In particular, when* $b = n^{2/3}$, *SMART achieves an* $\varepsilon > 0$ *accurate solution with, on average,* $O(n + \kappa n^{2/3}\log(1/\varepsilon))$ *gradient evaluations,* $O(\kappa\log(1/\varepsilon))$ $\mathbf{prox}_{\gamma r_2}$*-proximal operator evaluations,* $O(\kappa n^{2/3}\log(1/\varepsilon))$ *function evaluations, and* $O((\kappa/n^{1/3})\log(1/\varepsilon))$ $\mathbf{prox}_{\tau r_1}$ *evaluations.*

4. Numerics

In this section we perform trimmed model fitting (i.e., we solve (1) with a regularizer) on three models/datasets:

1. recognizing hand-written digits (0-9) with multinomial classification on the **mnist** dataset (LeCun et al., 1998);
2. trimmed principal component analysis, using the **US Judges** dataset provided in R (R Development Core Team, 2008);
3. robust homography estimation using interest point matching.

The latter two applications are formulated using nonconvex constraints. Plots for figures 3 and 4 were generated with Matplotlib (Hunter, 2007).

4.1 Multi-class classification

The **mnist** training dataset contains 60000 pictures of hand-written digits between 0-9. We model automated digit recognition as a multi-class classification problem with $K = 10$ classes. We briefly review multinomial logistic regression to align (1) with our current formulation.

Formulation: We are given n data pairs (v_i, y_i), where $v_i \in \mathbb{R}^p$ are training features, and $b_i \in \mathbb{R}^K$ are 'one-hot' training labels. If the ith example belongs to the jth class, then $y_i = e_j$, the jth standard unit vector.

The decision variable is a matrix $X \in \mathbb{R}^{p\times K}$ and each column x_j of X defines a linear classifier. The soft-max loss is a standard objective for selecting the best fitting classifier out of a given set:

$$f_i(X) = -\log\left(\frac{\exp(v_i^T X y_i)}{\sum_{j=1}^K \exp(\langle v_i, x_j\rangle)}\right) = \log\left(\sum_{j=1}^K \exp(\langle v_i, x_j\rangle)\right) - v_i^T X y_i.$$

Define the log-sum-exp(LSE) function by $\mathrm{LSE}(z) = \log\left(\sum_j \exp(z_j)\right)$. The LSE function is smooth, with gradient and hessian given by

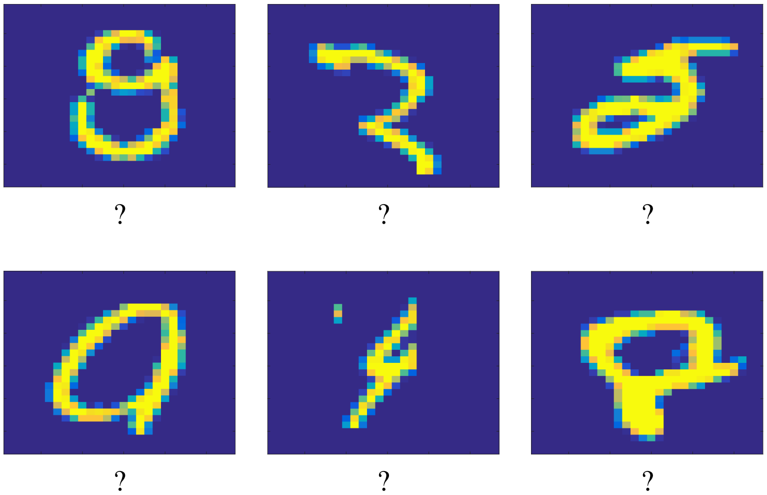
$$\nabla_{x_j} f_i = \left(\frac{\exp(\langle v_i, x_j\rangle)}{\exp(\mathrm{LSE}(Xv_i))} - y_{ij}\right) v_i.$$
$$\nabla^2 \mathrm{LSE}(z) = \frac{1}{\exp(\mathrm{LSE}(z))}\mathrm{diag}(\exp(z)) - \frac{1}{\exp(\mathrm{LSE}(z))^2}\exp(z)\exp(z)^T,$$

and so the Lipschitz constant of $\nabla\mathrm{LSE}$ is no greater than 2.

The trimmed (regularized) multiclass problem is given by

$$\min_X \min_{w\in\Delta^m} \frac{1}{n}\sum_{i=1}^n w_i\left(\mathrm{LSE}(Xv_i) - v_i^T X y_i\right) + R(X). \tag{7}$$

For simplicity, we use $R(x) = \frac{\lambda}{2n}\|x\|^2$, where n is the number of examples.


? ? ?

? ? ?

Figure 2: Six of the outliers found in the original MNIST dataset using trimming. Can you guess what the labels are? (They are given in the text describing the experiment).

Experiments: We use $\lambda = 0.01$ consistently for all experiments. We first set $m = 0.998n$, to find outliers in the actual **mnist** dataset. The results are shown in Figure 2. The labels in the top row are $9, 3, 5$ and labels for the bottom row are $9, 4, 8$. Some labels simply appear to be flipped; for example, the top left figure is possibly an '8' rather than a '9'. Some images contain features that do not belong; for example, there is an extra loop in the 5 (top right) and a missing loop in the 8 (bottom right). Studying outliers, once they are detected, can give interesting insights into the learning example.

Messily written digits plague **mnist** training and test sets, so we should not expect that removing potential outliers from the training set improves classification performance on the test set. However, when we maliciously contaminate the **mnist** training set by shifting a large portion of the labels by 1 (modulo 9), trimming accuracy degrades only slightly, while the standard approach fails dramatically.

We show the effects of malicious contamination in Table 2. For the trimmed formulation, we always over-estimate the proportion of outliers by 10%. Then, we evaluate the predictive accuracy of the trimmed and standard approaches on the test set. We also evaluate how well each method detects outliers.

For the standard approach, we fit the untrimmed LSE model and then label as outliers the data points which obtain the $n - h$ largest objective values. This approach is standard in regression. For the trimmed method, the outliers are determined by the zero-set of the w vector.

The results are shown in Table 2. While the trimmed formulation (solved with SMART) degrades only slightly with between 10%- 40% systematic contamination, the standard approach degrades much more rapidly. Even with 40% mislabeled data, SMART is able to identify more than 95% of the outliers that we maliciously injected.

When the proportion of systematic errors reaches 50%, both methods degrade rapidly. This is not surprising: when 50% of labeled data is both wrong and mutually consistent, we are just as likely to find the incorrect model.

Outliers	**0%**	**10%**	**20%**	**30%**	**40%**	**50%**
LSE-Accuracy	92.28	89.2	85.3	78.8	65.4	44.9
LSE-Detection	—	90.8	90.4	82.4	71.8	61.0
LSE-False-Pos	—	11.5	14.9	21.8	35.5	59.0
SMART-Accuracy	91.2	90.7	89.9	89.0	86.8	43.7
SMART-Detection	—	99.6	99.1	98.2	96.8	61.7
SMART-False-Pos	—	11.4	12.7	16.4	19.5	58.6

Table 2: Accuracy, detection, and false positive rate for standard (LSE) approach and trimmed approach. The approximate number of outliers, required by SMART, is consistently over-estimated by 10% to reflect a realistic application of the method. The first column shows that over-estimation of outliers by 10% even in the nominal case carries only a 1% cost in terms of predictive accuracy.

Performance comparison with PALM and SG. In Figure 3, we compare SMART to PALM (Bolte et al., 2014) and SG. In all of our experiments, we manually found the best stepsizes γ and τ for PALM, SMART, and SG. We chose SMART's batch size to be $b := \lceil n^{2/3} \rceil = 1533$. For a fair comparison, we ran SG with a minibatch of the same size. Because (7) is nonsmooth and nonconvex, there is no available method to determine the global minimizer (w^*, x^*) of F. As a proxy for $F(w^*, x^*)$, we ran SMART multiple times, for many iterations, and chose the lowest achieved objective value. We found that although PALM and SG are competitive with SMART during the first few passes through the dataset, their performance quickly stagnates, possibly due to finding spurious stationary points.

4.2 Trimmed Principal Component Analysis (T-PCA)

For a given matrix $A \in \mathbb{R}^{m \times n}$, we can analyze its principal linear components by finding, in the least squares sense, the best rank k approximation to A. The principal linear components of A can be found through its singular value decomposition

$$A = UDV^T, \tag{8}$$

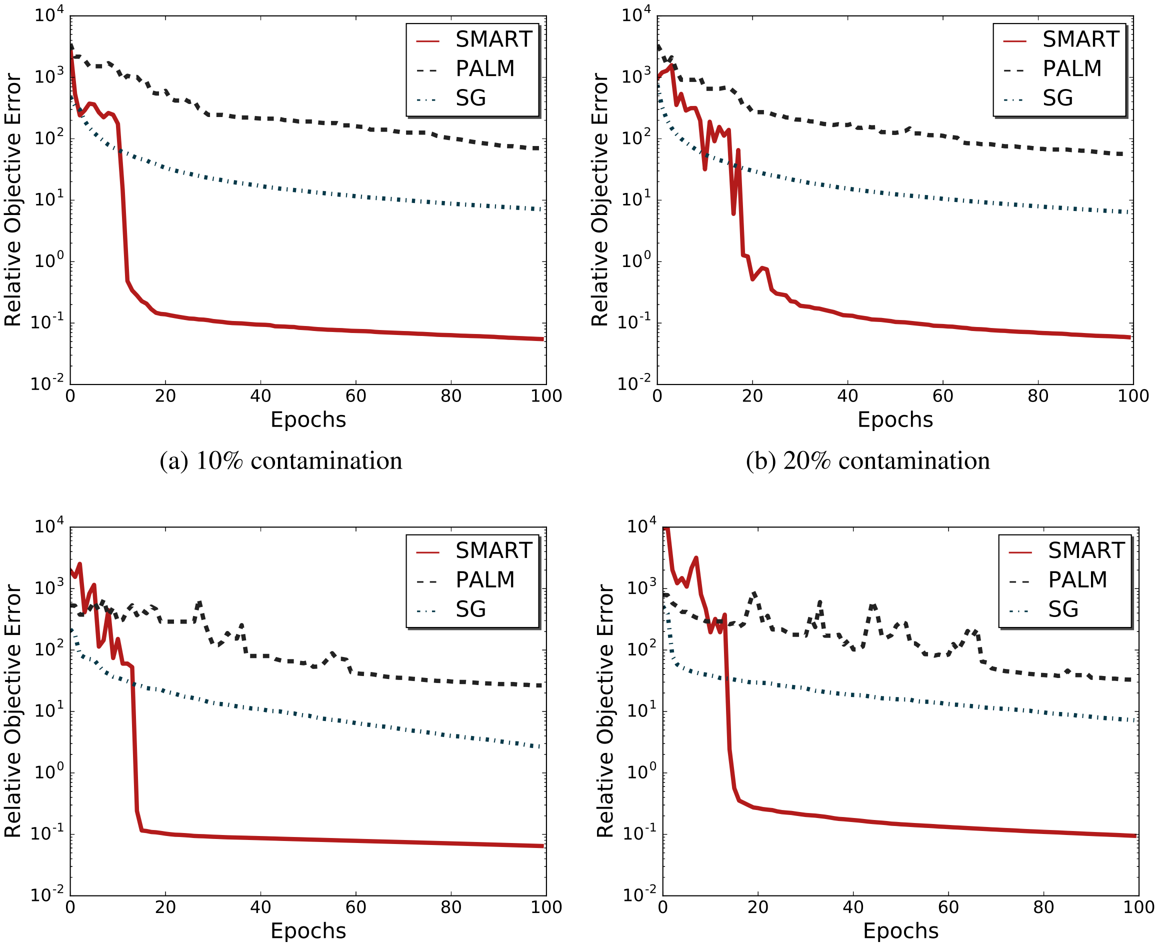
where $U \in \mathbb{R}^{m \times k}$ and $V \in \mathbb{R}^{m \times k}$ are orthogonal matrices, while $D \in \mathbb{R}^{k \times k}$ is diagonal with non-negative entries. The columns of the matrix $X = UD$ are the *principal components* of A and V is their corresponding *loadings*. This process of finding U, V, D and X is called *Principal Component Analysis* (PCA).

Formulation. It is well known that the matrix U in PCA minimizes

$$\min_{U \in \mathcal{O}^{m \times k}} \frac{1}{2} \|(I - UU^T)A\|^2, \tag{9}$$

where $\mathcal{O}^{m \times k}$ is the set of $m \times k$ matrices with orthonormal columns. Trimmed Principal Component Analysis (T-PCA) seeks to find such a U while simultaneously removing the influence of potentially contaminated columns a_i of A:

$$\min_{w \in \Delta^h, U \in \mathcal{O}^{m \times k}} \frac{1}{n} \sum_{i=1}^{n} \frac{w_i}{2} \left\| (I - UU^T) a_i \right\|^2. \tag{10}$$



(a) 10% contamination

(b) 20% contamination

(c) 30% contamination

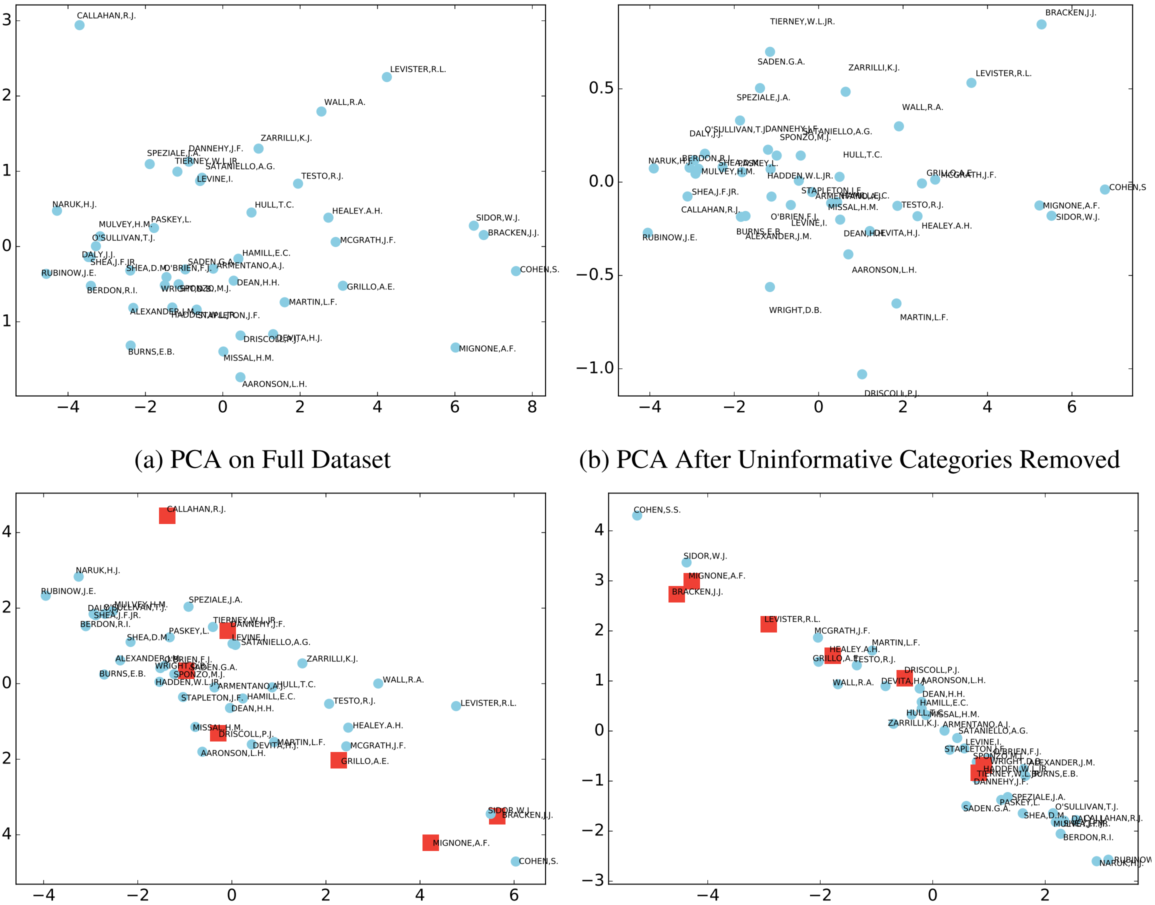
(d) 40% contamination

Figure 3: Comparison of SMART (SVRG variant in Corollary 3), PALM (Bolte et al., 2014), and a SG with minibatching (of size $n^{2/3}$) in terms of relative objective error $(F(w^k, x^k) - F(w^*, x^*))/F(w^*, x^*)$, which is not computed at every iteration, but only at the start of each epoch (i.e., after each full pass through all 60000 datapoints). In each of the four subplots, we maliciously contaminated a certain portion of the training labels by adding to each the number 1 modulo 9. Then, we solve the trimmed formulation (7). In all of the experiements, we overestimate the number of contaminated points by 10%. We show the classification performance of the SMART estimator in Table 2.

Note that $U \in \mathcal{O}^{m\times k}$, implies that

$$\left\|(I - UU^T)a_i\right\|^2 = \|a_i\|^2 - \|U^T a_i\|^2.$$

Thus, the PCA loss function is the sum of concave functions (each with a Lipschitz continuous derivative), while the regularizer r_2 is the indicator function of the orthogonal manifold $\mathcal{O}^{m\times k}$. When combined with trimming, PCA is highly nonconvex. Nevertheless, by Theorem 1, SMART will



(a) PCA on Full Dataset

(b) PCA After Uninformative Categories Removed

(c) 20% Trimming on Full Dataset

(d) 20% Trimming After Uninformative Categories Removed

Figure 4: PCA and trimmed PCA on the **US Judges** dataset. The left column depicts PCA and trimmed PCA on the full data matrix $A \in \mathbb{R}^{12\times 43}$ (i.e., the datasets $U_{A,\text{PCA}}^T A \in \mathbb{R}^{2\times 43}$ and $U_{A,\text{T-PCA}}^T A \in \mathbb{R}^{2\times 43}$, where $U_{A,\text{PCA}}$ and $U_{A,\text{T-PCA}}$ are found in (9) and (10), respectively), while the right column depicts PCA and trimmed PCA on the reduced data matrix $B \in \mathbb{R}^{8\times 43}$ (i.e., the datasets $U_{B,\text{PCA}}^T B \in \mathbb{R}^{2\times 43}$ and $U_{B,\text{T-PCA}}^T B \in \mathbb{R}^{2\times 43}$, where $U_{B,\text{PCA}}$ and $U_{B,\text{T-PCA}}$ are found in (9) and (10), respectively); see the text for a description of these matrices.

converge (almost surely) when applied to this problem because the iterates U^k lie in the bounded set of orthogonal matrices.

Although it may seem that computing $\mathbf{prox}_{\gamma r_2} = P_{\mathcal{O}^{m\times k}}$ dominates the cost of SMART on the trimmed-PCA problem, in reality the condition $k \approx b$ ensures that the costs of gradient and projection steps are balanced. Indeed, each batch gradient with $b = n^{2/3}$ samples requires $O(kmn^{2/3})$ arithmetic operations, while each U-projection requires only $O(mk^2)$ operations.

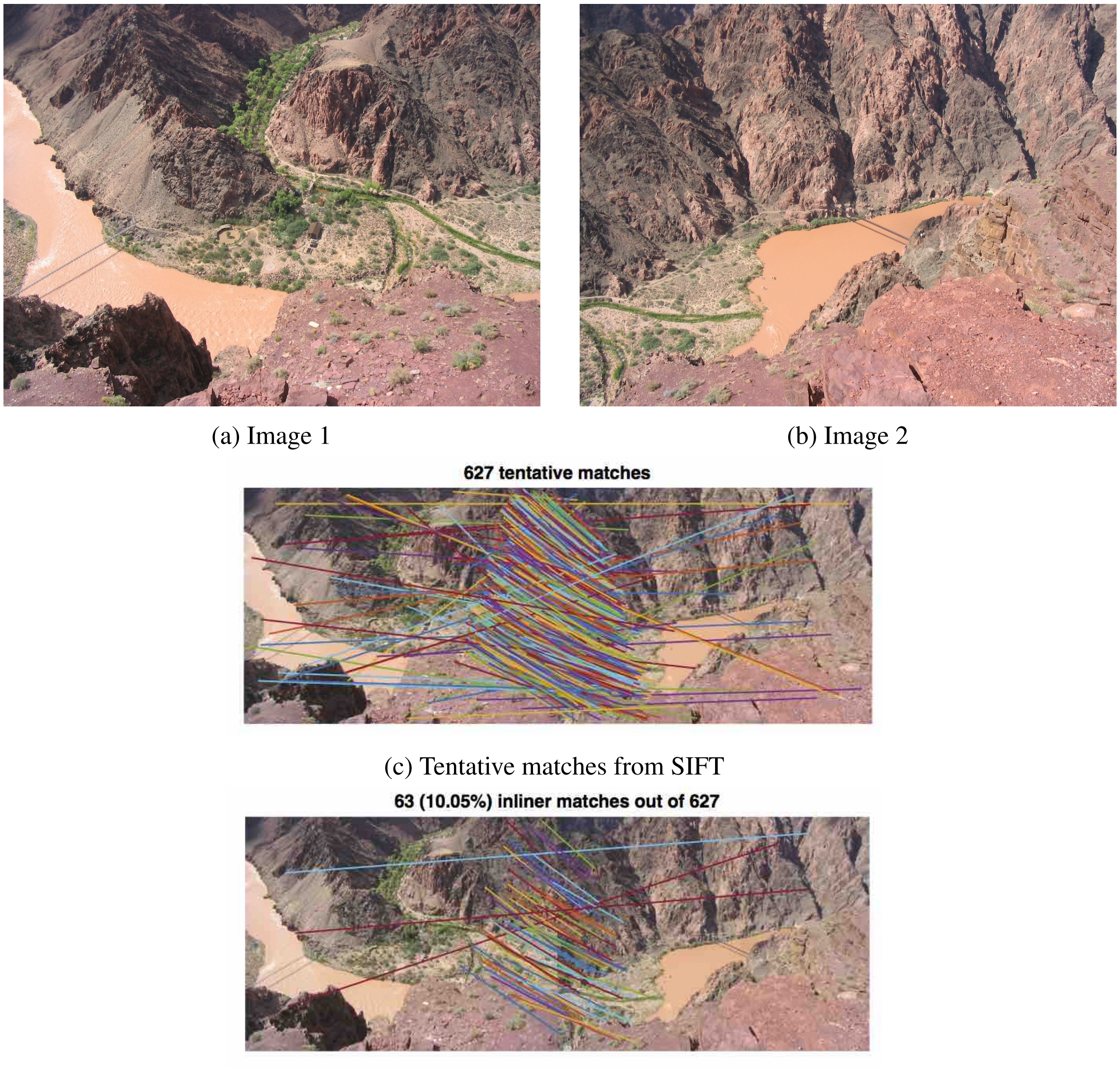
Experiments We used the **US judges** datset to test trimmed-PCA. This datasets collects lawyers' ratings of 43 different judges based on 12 numeric variables: number of contacts of lawyer with judge (`CONT`), judicial integrity (`INTG`), demeanor (`DMNR`), diligence (`DILG`), case flow managing (`CFMG`), prompt decisions (`DECI`), preparation for trial (`PREP`), familiarity with law (`FAMI`), sound oral rulings (`ORAL`), sound written rulings (`WRIT`), physical ability (`PHYS`), and worthy of retention (`RTEN`). We are interested in ranking the judges by quality.

After standardizing the matrix $A \in \mathbb{R}^{12\times 43}$ (by ensuring each row had mean zero), we computed PCA of this dataset (Figure 4a), with $k = 2$. As evident in the plot, the data lacks directionality, which possibly means we have chosen k to be too small.

Next we used SMART to compute 20%-trimmed PCA on A (Figure 4c, discovered outliers plotted as red squares). After trimming 20% of the dataset, it exhibited much greater directionality. In particular, the judges in the bottom right corner of Figure 4c were rated poorly across all dimensions, while the judges in the upper left were rated highly across all dimensions. The median ratings across the dataset were as follows: `CONT 7.3, INTG 8.1, DMNR 7.7, DILG 7.8, CFMG 7.6, DECI 7.7, PREP 7.7, FAMI 7.6, ORAL 7.5, WRIT 7.6, PHYS 8.1, RTEN 7.8`. There was no clear pattern among the outlying judges. Some were rated especially high, for example, `CALLAHAN, R.J.` was rated `CONT 10.6, INTG 9, DMNR 8.9, DILG 8.7, CFMG 8.5, DECI 8.5, PREP 8.5, FAMI 8.5, ORAL 8.6, WRIT 8.4, PHYS 9.1, RTEN 9`, some were rated especially low, for example, `BRACKEN, J.J.` was rated `CONT 7.3, INTG 6.4, DMNR 4.3, DILG 6.5, CFMG 6, DECI 6.2, PREP 5.7, FAMI 5.7, ORAL 5.1, WRIT 5.3, PHYS 5.5, RTEN 4.8`, while others were rated close to the median in some respects and distant in others, for example, `DRISCOLL, P.J.` was rated `CONT 6.7, INTG 8.6, DMNR 8.2, DILG 6.8, CFMG 6.9, DECI 6.6, PREP 7.1, FAMI 7.3, ORAL 7.2, WRIT 7.2, PHYS 8.1, RTEN 7.7`.

We hypothesized that some of the 12 variables were uninformative for predicting the quality of a judge. For example, it is not clear how `CONT` relates to quality because it is controled by the judge, but may depend on the trial. Thus, we used SMART to compute 60%-trimmed PCA on the transposed matrix $A^T \in \mathbb{R}^{43\times 12}$ and discovered the outlying categories `CONT`, `DMNR`, `INTG`, and `PHYS`. We removed these variables from the dataset, which resulted in a reduced data matrix $B \in \mathbb{R}^{8\times 43}$. Then we performed PCA on this new data matrix B (Figure 4b). Interestingly, some of the outliers found by 20%-trimmed PCA on A, for example `BRACKEN, J.J` and `DRISCOLL, P.J.`, were removed from the center of the point cloud, making them easier to spot visually, while others no longer appeared to be outliers, for example, `CALLAHAN, R.J.`

The point cloud produced by standard PCA still lacked clear directionality. Thus, we used SMART to compute 20%-trimmed PCA on B (Figure 4d, discovered outliers plotted as red squares). Figure 4d shows that trimmed PCA now found a clear linear component of the data: the judges in the upper left hand are poorly rated, the judges in the middle of the figure are near the median, and the judges in the bottom right are highly rated. Compared to 20%-trimmed PCA on A, some of the outliers persist, for example, `BRACKEN, J.J` and `DRISCOLL, P.J.`, while others cease to be outliers, for example, `CALLAHAN, R.J.` and `DANNEHY, J.F.` One hypothesis for why `DRISCOLL, P.J.` persists as an outlier is that he or she was rated low with respect to `DILG`, `CFMG, DECI` and `PREP`, but is still considered worthy of retention. One hypothesis for why `CALLAHAN, R.J.` was an outlier with respect to A and not with respect to B is that he or she received an extremely high rating for `CONT`, 10.6, while the mean and median for these ratings were 7.4372 and 7.3, respectively.



(a) Image 1

(b) Image 2

(c) Tentative matches from SIFT

(d) Best 10% matches found by SMART

Figure 5: Panels (a) and (b) show two related images with some overlapping features. Panel (c) shows tentative point correspondences discovered by SIFT. There are many spurious incorrect matches. Panel (d) shows the trimmed correspondences obtained using SMART. All images and feature matches were generated with VLFeat (Vedaldi and Fulkerson, 2008).

4.3 Robust homography estimation

Two images of the same scene, taken by a pin-hole camera, are related by a *homography* (see e.g. Hartley and Zisserman (2003); Ma et al. (2012)). There exists a unique matrix $H \in \mathbb{R}^{3\times 3}$ with $\|H\|_F^2 = 1$ so that given any corresponding pair of points (u_1, v_1) in image 1 and (u_2, v_2) in image

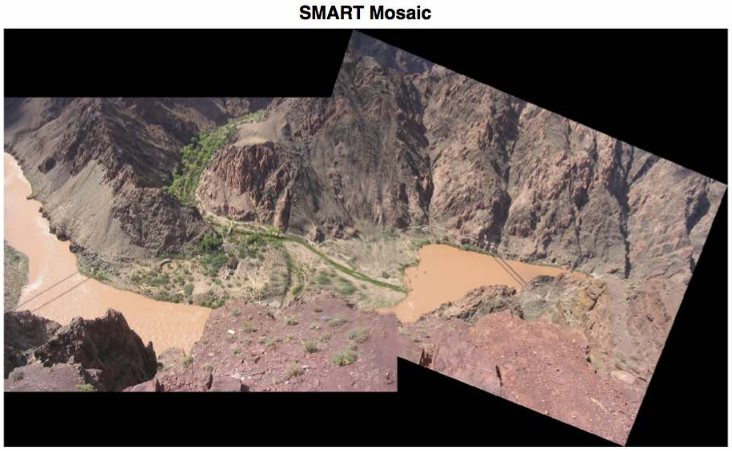


Figure 6: Final mosaic for images in Figure 5 obtained using the homography estimated by (13).

2, we have

$$H \begin{bmatrix} u_1 \\ v_1 \\ 1 \end{bmatrix} = \begin{bmatrix} u_2 \\ v_2 \\ 1 \end{bmatrix}. \tag{11}$$

Given a set of point correspondences, we can determine H. Arranging corresponding sets of points into matrices B_1 and B_2, we can solve

$$\min_{\|H\|_F = 1} \|HB_1 - B_2\|_F^2. \tag{12}$$

The solution of (12) can be characterized by $A\text{vec}(H) = 0$ (Hartley and Zisserman, 2003), with

$$A = \begin{bmatrix} 0 & 0 & 0 & -u & -v & -1 & v'u & v'v & v' \\ u & v & 1 & 0 & 0 & 0 & -u'u & -u'v & -u' \end{bmatrix},$$

and 4 points (with no 3 collinear) are enough to determine H. Given a perfect set of 4 point correspondences, the solution $\text{vec}(H)$ is immediately obtained from the right singular vector of A (with singular value 0), foregoing any iterative algorithm to required to solve (12). This approach is known as direct linear transformation (DLT) (Abdel-Aaziz and Karara, 1971).

The main challenge for homography estimation is finding a correct set of point correspondences. Potential point correspondences are generated with two steps. First, each image is scanned for visually distinctive points. Those points deemed distinctive are assigned a vector (typically a 128 dimensional scale-invariant feature transform (SIFT) (Lowe, 1999) descriptor) that summarizes the neighborhood of the interest point. Second, by comparing descriptors between the images (typically with a nearest neighbors test) potential correspondences are generated between distinctive points.

After potential correspondences are generated, the random sample consensus (RANSAC) algorithm (Fischler and Bolles, 1981) is used to remove erroneous correspondences. To do this, RANSAC repeatedly selects a set of 4 points correspondences (uniformly at random), fits h using the DLT procedure, and then estimates a consensus set, i.e. a set of point pairs (m_1, m_2) whose errors $\|Hm_1 - m_2\|$ are smaller than a pre-defined threshold. Once the consensus set is large enough, the algorithm stops.

Formulation. Given n point correspondences, rewriting (12) as a sum over data points, and introducing weights, we solve

$$\min_{w \in \Delta^h, \|H\|_F = 1} \sum_{i=1}^{n} w_i \|Hb_{1,i} - b_{2,i}\|^2. \tag{13}$$

Note that this formulation includes the nonconvex constraint $\|H\|_F = 1$. We take the predicted number of inliers to be a small proportion of the data, say, 10% or 20%.

Experiments. We use (13) to stitch together two overlapping images (shown in Figure 5). In our experiment, there are 627 point correspondences between the images (shown in Figure 5c). Many of these correspondences are spurious. We trim away 90% of the data using the SMART formulation (13), leaving only the correspondences shown in Figure 5d. After solving (13), we do a refinement step to estimate the final homography. We select the four best fitting correspondences (i.e., those with lowest objective values) and apply the DLT method as detailed above.

Although SMART recovers a good mosaic, similar mosaics can also be recovered by RANSAC. However, for larger scale bundle adjustment problems, in which multiple images of the same scene are used to estimate several interconnected homographies, RANSAC becomes prohibitively slow. We expect SMART to perform well on these problems, but we leave them to future work.

5. Conclusion

We introduced the SMART algorithm for solving the highly nonconvex, nonsmooth problem (2), which was motivated by the nonconvex trimming problem (1). SMART is the first stochastic gradient algorithm for fully nonconvex optimization that provably converges. Moreover, SMART scales better, by a factor of $n^{1/3}$, than all competing full gradient methods. In spite of the nonsmooth, nonconvex nature of (2), we showed that SMART converges quickly, performs meaningful inference on contaminated datasets, and reliably detects outliers.

References

Y. Abdel-Aaziz and H. Karara. Direct linear transformation from comparator co, ordinates into object space coordinates in close range photogrammetry. In *ASP Symposium on close range photogrammetry, Fall Church*, 1971.

A. Alfons, C. Croux, S. 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.

A. Aravkin, D. Drusvyatskiy, and T. van Leeuwen. Variable projection without smoothness. *arXiv preprint arXiv:1601.05011*, 2016.

H. Attouch, J. Bolte, and B. F. Svaiter. Convergence of descent methods for semi-algebraic and tame problems: proximal algorithms, forward–backward splitting, and regularized gauss–seidel methods. *Mathematical Programming*, 137(1):91–129, 2013. ISSN 1436-4646.

H. H. Bauschke and P. L. Combettes. *Convex analysis and monotone operator theory in Hilbert spaces*. Springer Science & Business Media, 2011.

J. Bolte, A. Daniilidis, and A. Lewis. The Łojasiewicz Inequality for Nonsmooth Subanalytic Functions with Applications to Subgradient Dynamical Systems. *SIAM Journal on Optimization*, 17(4):1205–1223, 2007a. doi: 10.1137/050644641.

J. Bolte, A. Daniilidis, A. Lewis, and M. Shiota. Clarke subgradients of stratifiable functions. *SIAM Journal on Optimization*, 18(2):556–572, 2007b.

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

D. Davis. The asynchronous palm algorithm for nonsmooth nonconvex problems. *arXiv preprint arXiv:1604.00526*, 2016a.

D. Davis. Smart: The stochastic monotone aggregated root-finding algorithm. *arXiv preprint arXiv:1601.00698*, 2016b.

D. Davis, B. Edmunds, and M. Udell. The sound of apalm clapping: Faster nonsmooth nonconvex optimization with stochastic asynchronous palm. *NIPS*, 2016.

A. Defazio, F. Bach, and S. Lacoste-Julien. SAGA: A Fast Incremental Gradient Method With Support for Non-Strongly Convex Composite Objectives. In *Advances in Neural Information Processing Systems*, pages 1646–1654, 2014.

D. Drusvyatskiy and A. S. Lewis. Error bounds, quadratic growth, and linear convergence of proximal methods. *arXiv preprint arXiv:1602.06661*, 2016.

M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6): 381–395, 1981.

S. Ghadimi, G. Lan, and H. Zhang. Mini-batch stochastic approximation methods for nonconvex stochastic composite optimization. *Mathematical Programming*, 155(1-2):267–305, 2016.

R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

P. J. Huber. *Robust Statistics*. John Wiley and Sons, 2004.

J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing In Science & Engineering*, 9(3): 90–95, 2007. doi: 10.1109/MCSE.2007.55.

K. L. Lange, R. J. A. Little, and J. M. G. Taylor. Robust Statistical Modeling Using the t Distribution. *Journal of the American Statistical Association*, 84:881–896, 1989.

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

D. G. Lowe. Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, pages 1150–1157. Ieee, 1999.

Y. Ma, S. Soatto, J. Kosecka, and S. S. Sastry. *An invitation to 3-d vision: from images to geometric models*, volume 26. Springer Science & Business Media, 2012.

R. A. Maronna, D. Martin, and Yohai. *Robust Statistics*. Wiley Series in Probability and Statistics. Wiley, 2006.

D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu. On the least trimmed squares estimator. *Algorithmica*, 69(1):148–183, 2014.

Y. Nesterov. *Introductory Lectures on Convex Optimization : A Basic Course*. Applied optimization. Kluwer Academic Publ., Boston, Dordrecht, London, 2004. ISBN 1-4020-7553-7.

N. M. Neykov and C. H. Müller. Breakdown point and computation of trimmed likelihood estimators in generalized linear models. In *Developments in robust statistics*, pages 277–286. Springer, 2003.

R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2008.

S. J. Reddi, S. Sra, B. Poczos, and A. Smola. Fast stochastic methods for nonsmooth nonconvex optimization. *arXiv preprint arXiv:1605.06900*, 2016.

R. T. Rockafellar and R. J. B. Wets. *Variational Analysis*, volume 317. Springer, 1998.

P. J. Rousseeuw. Least median of squares regression. *Journal of the American statistical association*, 79(388):871–880, 1984.

P. J. Rousseeuw. Multivariate estimation with high breakdown point. *Mathematical statistics and applications*, 8:283–297, 1985.

P. J. Rousseeuw and K. Van Driessen. Computing lts regression for large data sets. *Data mining and knowledge discovery*, 12(1):29–45, 2006.

D. Ruppert and R. J. Carroll. Trimmed least squares estimation in the linear model. *Journal of the American Statistical Association*, 75(372):828–838, 1980.

A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms. `http://www.vlfeat.org/`, 2008.

L. Xiao and T. Zhang. A Proximal Stochastic Gradient Method with Progressive Variance Reduction. *SIAM Journal on Optimization*, 24(4):2057–2075, 2014.

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.

Appendix A. Proof of Theorem 1

Notation. We will often repeat the following terms

- **Conditional expectation** $\mathbb{E}_k$ For every $k \in \mathbb{N}$, and every random variable X, we let
$$\mathbb{E}_k[X] = \mathbb{E}[X \mid \mathcal{F}_k],$$
where $\mathcal{F}_k$ is defined as in Assumption 1.

- **Stochastic Gradient Estimator.** For all $k \in \{0, \ldots, T-1\}$, we define an $\mathcal{H}$-valued random variables v^k with components
$$v^k := \frac{1}{b}\sum_{i \in I_k}(w_i^k \nabla f_i(x^k) - y_i^k) + \frac{1}{n}\sum_{i=1}^n y_i^k.$$

- **Full Update.** For all $k \in \{0, \ldots T-1\}$, we define a vector $\overline{z}^{k+1} \in \mathcal{H}$ componentwise:
$$\overline{z}^{k+1} = \mathbf{prox}_{\gamma r_2}\left(x^k - \gamma v^k\right).$$

- **The β_i Factors.** Set
$$\beta_i := \sqrt{1-\rho_i}\left(\frac{1}{\sqrt{q'}} - \sqrt{1-\rho_i}\right).$$

- **The a Factor.** Set
$$a = \gamma L\sqrt{\frac{1}{n}\sum_{i=1}^n \frac{q'(1+\epsilon_0)(B_i)^2}{2b\left(1-\sqrt{q'(1-\rho_i)}\right)^2}}.$$

- **The α_i Factors.** We let
$$\alpha_i := \frac{q'\gamma(1+\epsilon_0)}{2ab}\sum_{t=0}^{\infty}\left[q'(1+\beta_i)(1-\rho_i)\right]^t = \frac{q'\gamma(1+\epsilon_0)}{2ab\left[1-q'(1+\beta_i)(1-\rho_i)\right]}.$$
We use the property that
$$\alpha_i = \alpha_i q'(1+\beta_i)(1-\rho_i) + \frac{q'\gamma(1+\epsilon_0)}{2ab}.$$

Parts 1 and 2. The supermartingale convergence theorem is our hammer for nailing down the effect of randomness in T-SMART:

Theorem 7 (Supermartingale Convergence Theorem) *Let $(\Omega, \mathcal{F}, P)$ be a probability space. Let $\mathfrak{F} := \{\mathcal{F}_k\}_{k\in\mathbb{N}}$ be an increasing sequence of sub σ-algebras of $\mathcal{F}$ such that $\mathcal{F}_k \subseteq \mathcal{F}_{k+1}$. Let $\{X_k\}_{k\in\mathbb{N}}$ and $\{Y_k\}_{k\in\mathbb{N}}$ be sequences of $[\xi, \infty)$-valued and $[0,\infty)$-valued random variables, respectively, such that for all $k \in \mathbb{N}$, X_k and Y_k are $\mathcal{F}_k$ measurable, and*
$$(\forall k \in \mathbb{N}) \qquad \mathbb{E}[X_{k+1} \mid \mathcal{F}_k] + Y_k \leq X_k. \tag{14}$$

Then $\sum_{k=0}^{\infty} Y_k < \infty$ a. s. and X_k a. s. converges to a $[\xi, \infty)$-valued random variable.

In this proof, we show that (14) holds for the random variables[5]

$$\begin{aligned}(\forall k \in \mathbb{N})\, X_k &= F(w^k, x^k) + \frac{1}{n}\sum_{i=1}^n \alpha_i \left\| w_i^k \nabla f_i(x^k) - y_i^k \right\|^2 . \\ Y_k &= \frac{q^{'}\gamma}{2\eta}\|\overline{x}^{k+1} - x^k\|^2 + \frac{q}{2\tau}\|\overline{w}^{k+1} - w^k\|^2 \\ &\quad + \frac{\epsilon_0 q^{'}\gamma}{2abn}\sum_{i=1}^n \|w_i^k \nabla f_i(x^k) - y_i^k\|^2 \end{aligned} \tag{15}$$

Then we apply Theorem 7 to show that $\sum_{k=0}^\infty Y_k < \infty$ a. s. and X_k a. s. converges to a $[F(w^*, x^*), \infty)$-valued random variable X_*.

Thus, there exists a full measure subset $\widetilde{\Omega} \subseteq \Omega$ such that the following hold: For all $\omega \in \widetilde{\Omega}$, the sequence $\{(w^k(\omega), x^k(\omega))\}_{k\in\mathbb{N}}$ is bounded and

1. Because $X_k \to X_*$ a. s. and $n^{-1}\sum_{i=1}^n \left\| w_i^k \nabla f_i(x^k) - y_i^k \right\| \to 0$ as $k \to \infty$, we have $F(w^k(\omega), x^k(\omega)) \to X_*(\omega)$ as $k \to \infty$.

2. Because $\sum_{k=0}^\infty Y_k < \infty$ a. s., we have $\|\overline{x}^{k+1}(\omega) - x^k(\omega)\|^2 \to 0$ and $\|\overline{w}^{k+1}(\omega) - w^k(\omega)\|^2 \to 0$ as $k \to \infty$.

We use these limits to prove properties of convergent subsequences of T-SMART along the full measure set $\widetilde{\Omega}$.

Lemma 8 *Let $\omega \in \widetilde{\Omega}$. Suppose that there exists an increasing sequence of indices $\{k_l\}_{l\in\mathbb{N}} \subseteq \mathbb{N}$ with the property that $(\overline{w}^{k_l+1}(\omega), \overline{x}^{k_l+1}(\omega)) \to (\overline{w}, \overline{x})$. Then $F(\overline{w}, \overline{x}) = X_*(\omega)$, the limit holds $F(\overline{w}^{k_l+1}(\omega), \overline{x}^{k_l+1}(\omega)) \to X_*(\omega) = F(\overline{w}, \overline{x})$, and there exists $g^{k_l} \in \partial_L F(\overline{w}^{k_l+1}, \overline{x}^{k_l+1})$ such that $g^{k_l} \to 0$ as $l \to \infty$. Therefore, $0 \in \partial_L F(\overline{w}, \overline{x})$.*

Thus, Parts 1 and 2 follows as soon as we prove (14) for X_k and Y_k. It turns out that Part 3 also follows from (14).

Part 3. If we apply the law of total expectation to (14), we find that

$$\min_{t=0,\ldots,T} \mathbb{E}\left[Y_t\right] \leq \frac{1}{T}\sum_{t=0}^T \mathbb{E}\left[Y_t\right] \leq \frac{\mathbb{E}\left[X_0 - X_{T+1}\right]}{T} \leq \frac{\mathbb{E}\left[F(w^0, x^0) - F(w^*, x^*)\right]}{T}$$

because $\sum_{i=1}^n \alpha_i \left\| w_i^0 \nabla f_i(x^0) - y_i^0 \right\| = 0$ and $F(w^T, x^T) \geq F(w^*, x^*)$.

Lemmas Leading to a Proof of (14) The proof of (14) requires four lemmas, whose proofs we defer for a moment. Though similar, the first lemma does not follow from (Reddi et al., 2016, Lemma 2).

[5] The variable X_k is clearly $\mathcal{F}_k$-measurable. The variable Y_k is $\mathcal{F}_k$-measurable because of our assumptions on ζ_1 and ζ_2.

Lemma 9 (Sufficient Decrease) *For all $k \in \mathbb{N}$, we have*

$$\begin{aligned}
&\mathbb{E}_k\left[F(w^{k+1}, x^{k+1})\right] \\
&\le F(w^k, z^k) + q'\mathbb{E}_k\left[\langle \overline{z}^{k+1} - \overline{x}^{k+1}, \frac{1}{n}\sum_{i=1}^n w_i^k \nabla f_i(x^k) - v^k\rangle\right] - \frac{q}{2\tau}\|\overline{w}^{k+1} - w^k\|^2 \\
&+ q'\left[\frac{L}{n}\sum_{i=1}^n B_i - \frac{\eta - 1}{2\gamma}\right]\|\overline{x}^{k+1} - x^k\|^2 + q'\mathbb{E}_k\left[\left[\frac{L}{n}\sum_{i=1}^n \frac{|w_i^k|}{2} - \frac{1}{2\gamma}\right]\|\overline{z}^{k+1} - x^k\|^2\right] \\
&- \frac{\epsilon_0 q'\gamma}{2abn}\sum_{i=1}^n \|w_i^k \nabla f_i(x^k) - y_i^k\|^2.
\end{aligned}$$

Lemma 10 (Variance Bound) *For all $k \in \mathbb{N}$, we have*

$$\mathbb{E}_k\left[\left\|\frac{1}{n}\sum_{i=1}^n w_i^k \nabla f_i(x^k) - v^k\right\|^2\right] \le \frac{1}{bn}\sum_{i=1}^n \|w_i^k \nabla f_i(x^k) - y_i^k\|^2.$$

Lemma 11 (Dual Variable Recursion) *For all $k \in \mathbb{N}$, and $i \in \{1, \ldots, n\}$, we have*

$$\begin{aligned}
\mathbb{E}_k\left[\|w_i^{k+1}\nabla f_i(x^{k+1}) - y_i^{k+1}\|^2\right] &\le q'\left(1 + \frac{1 - \rho_i}{\beta_i}\right)(Lw_i^k)^2 \mathbb{E}_k\left[\|\overline{z}^{k+1} - x^k\|^2\right] \\
&+ q'(1 - \rho_i)(1 + \beta_i)\left\|w_i^k \nabla f_i(x^k) - y_i^k\right\|^2.
\end{aligned}$$

Lemma 12 (α_i bound) *The following bound holds:*

$$\frac{1}{n}\sum_{i=1}^n \left[\alpha_i (w_i^k L)^2 \left(1 + \frac{(1 - \rho_i)}{\beta_i}\right) + \frac{|w_i^k| L}{2}\right] \le \frac{1 - 2a}{2\gamma}.$$

Proof of (14) Using the variance bound, we bound the cross term from Lemma 9:

$$\begin{aligned}
&\mathbb{E}_k\left[\langle \overline{z}^{k+1} - \overline{x}^{k+1}, \frac{1}{n}\sum_{i=1}^n w_i^k \nabla f_i(x^k) - v^k\rangle\right] \\
&\le \mathbb{E}_k\left[\frac{a}{2\gamma}\|\overline{z}^{k+1} - \overline{x}^{k+1}\|^2 + \frac{\gamma}{2a}\left\|\frac{1}{n}\sum_{i=1}^n w_i^k \nabla f_i(x^k) - v^k\right\|^2\right] \\
&\le \mathbb{E}_k\left[\frac{a}{2\gamma}\|\overline{z}^{k+1} - \overline{x}^{k+1}\|^2\right] + \frac{\gamma}{2abn}\sum_{i=1}^n \|w_i^k \nabla f_i(x^k) - y_i^k\|^2 \\
&\le \mathbb{E}_k\left[\frac{a}{\gamma}\|\overline{z}^{k+1} - x^k\|^2\right] + \frac{a}{\gamma}\|\overline{x}^{k+1} - x^k\|^2 + \frac{\gamma}{2abn}\sum_{i=1}^n \|\nabla f_i(x^k) w_i^k - y_i^k\|^2,
\end{aligned}$$

where the first inequality follows from the Cauchy-Schwarz inequality and the bound $cd \le c^2 a/(2\gamma) + d^2\gamma/2a$, and the last inequality follows from the bound $\|c + d\|^2 \le 2\|c\|^2 + 2\|d\|^2$ and the $\mathcal{F}_k$-measurability of $\|\overline{x}^{k+1} - z^k\|^2$.

Thus, the cross term bound taken together with Lemma 9 yields

$$\begin{aligned}
&\mathbb{E}_k\left[F(w^{k+1}, x^{k+1})\right]\\
&\leq F(w^k, x^k) + q'\left[\frac{L}{n}\sum_{i=1}^n B_i - \frac{(\eta - 1 - 2a)}{2\gamma}\right]\|\overline{x}^{k+1} - x^k\|^2\\
&+ q'\mathbb{E}_k\left[\left[\frac{L}{n}\sum_{i=1}^n \frac{|w_i^k|}{2} - \frac{1-2a}{2\gamma}\right]\|\overline{z}^{k+1} - x^k\|^2\right] + \frac{q'\gamma}{2abn}\sum_{i=1}^n \|w_i^k \nabla f_i(w^k) - y_i^k\|^2.\\
&- \frac{q}{2\tau}\|\overline{w}^{k+1} - w^k\|^2.
\end{aligned}$$

Therefore,

$$\begin{aligned}
&\mathbb{E}_k\left[X_{k+1}\right]\\
&\leq F(w^k, x^k) + \frac{1}{n}\sum_{i=1}^n \mathbb{E}_k\left[\alpha_i\left\|w_i^{k+1}\nabla f_i(x^{k+1}) - y_i^{k+1}\right\|^2\right]\\
&+ q'\left[\frac{L}{n}\sum_{i=1}^n B_i - \frac{(\eta - 1 - 2a)}{2\gamma}\right]\|\overline{x}^{k+1} - x^k\|^2 - \frac{q}{2\tau}\|\overline{w}^{k+1} - w^k\|^2\\
&+ q'\mathbb{E}_k\left[\left[\frac{L}{n}\sum_{i=1}^n \frac{|w_i^k|}{2} - \frac{1-2a}{2\gamma}\right]\|\overline{z}^{k+1} - x^k\|^2\right] + \frac{q'\gamma}{2abn}\sum_{i=1}^n \|w_i^k \nabla f_i(x^k) - y_i^k\|^2.
\end{aligned}$$

Using the dual variable recursion bound, we find that

$$\begin{aligned}
&\frac{1}{n}\sum_{i=1}^n \mathbb{E}_k\left[\alpha_i\left\|w_i^{k+1}\nabla f_i(x^{k+1}) - y_i^{k+1}\right\|^2\right]\\
&\leq \mathbb{E}_k\left[\frac{1}{n}\sum_{i=1}^n \alpha_i q'(w_i^k L)^2\left[1 + \frac{(1-\rho_i)}{\beta_i}\right]\|\overline{z}^{k+1} - x^k\|^2\right]\\
&\qquad + \frac{1}{n}\sum_{i=1}^n \alpha_i q'(1+\beta_i)(1-\rho_i)\left\|w_i^k\nabla f_i(x^k) - y_i^k\right\|^2.
\end{aligned}$$

Thus,

$$
\begin{aligned}
&\mathbb{E}_k\left[X_{k+1}\right] \\
&\leq F(w^k, z^k) + \frac{1}{n}\sum_{i=1}^n \left[\alpha_i q'(1+\beta_i)(1-\rho_i) + \frac{\gamma q'(1+\epsilon_0)}{2ab}\right]\left\|w_i^k\nabla f_i(x^k) - y_i^k\right\|^2 \\
&\qquad + q'\mathbb{E}_k\left[\left[\frac{1}{n}\sum_{i=1}^n\left[\alpha_i(w_i^kL)^2\left(1+\frac{(1-\rho_i)}{\beta_i}\right)+\frac{w_i^kL}{2}\right] - \frac{1-2a}{2\gamma}\right]\|\overline{z}^{k+1}-x^k\|^2\right] \\
&\qquad + q'\left[\frac{L}{n}\sum_{i=1}^n B_i - \frac{(\eta-1-2a)}{2\gamma}\right]\|\overline{x}^{k+1}-x^k\|^2 \\
&\qquad - \frac{q}{2\tau}\|\overline{w}^{k+1}-w^k\|^2 - \frac{\epsilon_0 q'\gamma}{2abn}\sum_{i=1}^n\|w_i^k\nabla f_i(x^k)-y_i^k\|^2. \\
&\leq X_k - \frac{q'\gamma}{2\eta}\left\|\frac{\eta}{\gamma}\left(\overline{x}^{k+1}-x^k\right)\right\|^2 - \frac{q\tau}{2}\left\|\frac{1}{\tau}\left(\overline{w}^{k+1}-w^k\right)\right\|^2 \\
&\qquad - \frac{\epsilon_0 q'\gamma}{2abn}\sum_{i=1}^n\|w_i^k\nabla f_i(x^k)-y_i^k\|^2 \\
&\leq X_k - Y_k
\end{aligned}
\tag{16}
$$

where the final inequalities follow from Lemma 12, the definition of α_i, and the identities

$$
\eta = 2\left(1+2a+2\gamma\frac{L}{n}\sum_{i=1}^n B_i\right).
$$

$$
\implies \frac{\eta}{2\gamma} = \frac{(\eta-1-2a)}{2\gamma} - \frac{L}{n}\sum_{i=1}^n B_i.
$$

Proofs of the Lemmas.

Proof [of Lemma 8] We first prove that $F(\overline{w}^{k_l+1}(\omega), \overline{x}^{k_l+1}(\omega)) \to F(\overline{w}, \overline{x})$, then we construct the subgradients.

Because $\|\overline{x}^{k+1}(\omega) - x^k(\omega)\|^2 \to 0$ and $\|\overline{w}^{k+1}(\omega) - w^k(\omega)\|^2 \to 0$ as $k \to \infty$, it follows that $(w^{k_l}, x^{k_l}) \to (\overline{w}, \overline{x})$ as $l \to \infty$. Thus, by continuity, we have

$$
\lim_{l\to\infty}\frac{1}{n}\sum_{i=1}^n \overline{w}_i^{k_l+1}(\omega)f_i(\overline{x}^{k_l+1}(\omega)) = \frac{1}{n}\lim_{l\to\infty}\sum_{i=1}^n w_i^{k_l}(\omega)f_i(x^{k_l}(\omega)) = \frac{1}{n}\sum_{i=1}^n \overline{w}_i(\omega)f_i(\overline{x}(\omega)).
$$

Proving that $\lim_{l\to\infty}\left\{r_1(\overline{w}^{k_l+1}(\omega)) + r_2(\overline{x}^{k_l+1}(\omega))\right\} = \lim_{l\to\infty}\left\{r_1(w^{k_l}(\omega)) + r_2(x^{k_l}(\omega))\right\} = r_1(\overline{w})+r_2(\overline{x})$ is a little subtler because r_1 and r_2 are not continuous, but merely lower-semicontinuous.

Because $F(w^k(\omega), x^k(\omega)) \to X_*(\omega)$, we know the following limit exists:

$$
\lim_{l\to\infty}\left\{r_1(w^{k_l}(\omega)) + r_2(x^{k_l}(\omega))\right\} = X_*(\omega) - \frac{1}{n}\sum_{i=1}^n \overline{w}_i(\omega)f_i(\overline{x}(\omega)).
$$

Now we focus on proving that $r_1(\overline{w}^{k_l+1}(\omega)) + r_2(\overline{x}^{k_l+1}(\omega))$ has the same limit as $l \to \infty$.

First,

$$r_1(\overline{w}^{k_l+1}(\omega)) \le r_1(w^{k_l}(\omega)) + \frac{1}{n}\sum_{i=1}^n (w_i^{k_l}(\omega) - \overline{w}_i^{k_l+1}(\omega)) f_i(x^{k_l}(\omega)) - \frac{1}{2\tau}\|\overline{w}^{k_l+1}(\omega) - w^{k_l}(\omega)\|^2$$

$$r_2(\overline{x}^{k_l+1}(\omega)) \le r_2(x^{k_l}(\omega)) + \left\langle \frac{1}{n}\sum_{i=1}^n w_i^{k_l}(\omega)\nabla f_i(x^{k_l}(\omega)), x^{k_l}(\omega) - \overline{x}^{k_l+1}(\omega)\right\rangle$$
$$- \frac{\eta}{2\gamma}\|\overline{x}^{k_l+1}(\omega) - x^{k_l}(\omega)\|^2.$$

Taking $\liminf$ of both sides as $l \to \infty$, we find that

$$\limsup_{l\to\infty}\left\{r_1(\overline{w}^{k_l+1}(\omega)) + r_2(\overline{x}^{k_l+1}(\omega))\right\} \le \lim_{l\to\infty}\left\{r_1(w^{k_l}(\omega)) + r_2(x^{k_l}(\omega))\right\},$$

where we've implicitly used that $\left\{(x^{k_l}(\omega), w^{k_l}(\omega))\right\}_{l\in\mathbb{N}}$ is bounded.

Second, for all $k \in \mathbb{N}$, define

$$d(k,w) = \max\{\{t < k \mid j_t(\omega) = 1\} \cup \{-1\}\} \quad \text{and} \quad d(k,x) = \max\{\{t < k \mid j_t(\omega) = 2\} \cup \{-1\}\}.$$

Without loss of generality, we now assume that k_0 is large enough that $d(k_0, w) > 0$ and $d(k_0, x) > 0$.

$$\begin{aligned}
r_1(w^{k_l}(\omega)) &\le r_1(\overline{w}^{k_l+1}(\omega)) + \frac{1}{n}\sum_{i=1}^n (\overline{w}_i^{k_l+1}(\omega) - w_i^{k_l}(\omega)) f_i(x^{d(k_l,w)}(\omega)) \\
&\quad + \frac{1}{2\tau}\|\overline{w}^{k_l+1}(\omega) - w^{d(k_l,w)}(\omega)\|^2 - \frac{1}{2\tau}\|w^{k_l}(\omega) - w^{d(k_l,w)}(\omega)\|^2. \\
&= r_1(\overline{w}^{k_l+1}(\omega)) + \frac{1}{n}\sum_{i=1}^n (\overline{w}_i^{k_l+1}(\omega) - w_i^{k_l}(\omega)) f_i(x^{d(k_l,w)}(\omega)) \\
&\quad + \frac{1}{2\tau}\left[2\langle \overline{w}^{k_l+1}(\omega) - w^{d(k_l,w)}(\omega), \overline{w}^{k_l+1}(\omega) - w^{k_l}(\omega)\rangle - \|\overline{w}^{k_l+1}(\omega) - w^{k_l}(\omega)\|^2\right],
\end{aligned}$$

and similarly for r_2, we have

$$\begin{aligned}
r_2(x^{k_l}(\omega)) &\le r_2(\overline{x}^{k_l+1}(\omega)) + \langle v^{d(k_l,x)}(\omega), \overline{x}^{k_l+1}(\omega) - x^{k_l}(\omega)\rangle \\
&\quad + \frac{\eta}{2\gamma}\left[\langle \overline{x}^{k_l+1}(\omega) - x^{d(k_l,x)}(\omega), \overline{x}^{k_l+1}(\omega) - x^{k_l}(\omega)\rangle - \|\overline{x}^{k_l+1}(\omega) - x^{k_l}(\omega)\|^2\right].
\end{aligned}$$

Thus, because $w^{d(k_l,w)}(\omega), x^{d(k_l,x)}(\omega), v^{d(k_l,x)}(\omega)$ are all bounded, by taking $\liminf$ of both sides as $l \to \infty$, we find that

$$\lim_{l\to\infty}\left\{r_1(w^{k_l}(\omega)) + r_2(x^{k_l}(\omega))\right\} \le \liminf_{l\to\infty}\left\{r_1(\overline{w}^{k_l+1}(\omega)) + r_2(\overline{x}^{k_l+1}(\omega))\right\}.$$

Therefore, we've shown that $r_1(\overline{w}^{k_l+1}(\omega)) + r_2(\overline{x}^{k_l+1}(\omega))$ and $r_1(w^{k_l}(\omega)) + r_2(x^{k_l}(\omega))$ have the same limit at $l \to \infty$. Now we show that $\lim_{l\to\infty}\left\{r_1(w^{k_l}(\omega)) + r_2(x^{k_l}(\omega))\right\} = r_1(\overline{w}) + r_2(\overline{x})$.

By the lower-semicontinuity of r_1 and r_2, we have

$$\lim_{l\to\infty}\left\{r_1(w^{k_l}(\omega)) + r_2(x^{k_l}(\omega))\right\} \geq r_1(\overline{w}) + r_2(\overline{x}).$$

In addition, by the definition of $w^{k_l}(\omega)$ and $x^{k_l}(\omega))$ as proximal points, we have

$$\begin{aligned} r_1(w^{k_l}(\omega)) &\leq r_1(\overline{w}) + \frac{1}{n}\sum_{i=1}^n (\overline{w}_i(\omega) - w_i^{k_l}(\omega)) f_i(x^{d(k_l,j)}(\omega)) \\ &\quad + \frac{1}{2\tau}\left[\langle \overline{w} - w^{d(k_l,w)}(\omega), \overline{w} - w^{k_l}(\omega)\rangle - \|\overline{w} - w^{k_l}(\omega)\|^2\right] \\ r_2(x^{k_l}(\omega)) &\leq r_2(\overline{x}) + \langle v^{d(k_l,x)}(\omega), \overline{x} - x^{k_l}(\omega)\rangle \\ &\quad + \frac{\eta}{2\gamma}\left[\langle \overline{x} - x^{d(k_l,x)}(\omega), \overline{x} - x^{k_l}(\omega)\rangle - \|\overline{x} - x^{k_l}(\omega)\|^2\right]. \end{aligned}$$

Therefore, by arguments similar to those already employed above, we find that

$$\lim_{l\to\infty}\left\{r_1(w^{k_l}(\omega)) + r_2(x^{k_l}(\omega))\right\} \leq r_1(\overline{w}) + r_2(\overline{x}).$$

Thus, $\lim_{l\to\infty}\left\{r_1(w^{k_l}(\omega)) + r_2(x^{k_l}(\omega))\right\} = r_1(\overline{w}) + r_2(\overline{x})$

Therefore, by taking all these limits together we have shown that $\lim_{l\to\infty} F(\overline{w}^{k_l+1}, \overline{x}^{k_l+1}) = \lim_{l\to\infty} F(w^{k_l}, x^{k_l}) = F(\overline{w}, \overline{x})$. Now we construct the subgradient $g^{k_l} \in \partial_L F(\overline{w}^{k_l+1}, \overline{x}^{k_l+1})$.

By definition of the proximal operator, we have

$$\begin{aligned} \frac{1}{\tau}\left(w^{k_l} - \overline{w}^{k_l+1}\right) &\in \frac{1}{n}(f_1(x^{k_l}), \ldots, f_n(x^{k_l})) + \partial_L r_1(\overline{w}^{k_l+1}); \\ \frac{\eta}{\gamma}\left(x^{k_l} - \overline{x}^{k_l+1}\right) &\in \frac{1}{n}\sum_{i=1}^n w_i^{k_l}\nabla f_i(x^{k_l}) + \partial_L r_2(\overline{x}^{k_l+1}), \end{aligned}$$

Then we let

$$g^{k_l} = \begin{bmatrix} \frac{1}{\tau}\left(w^{k_l} - \overline{w}^{k_l+1}\right) + \frac{1}{n}(f_1(\overline{x}^{k_l+1}) - f_1(x^{k_l}), \ldots, f_n(\overline{x}^{k_l+1}) - f_n(x^{k_l})) \\ \frac{\eta}{\gamma}\left(x^{k_l} - \overline{x}^{k_l+1}\right) + \frac{1}{n}\sum_{i=1}^n \overline{w}_i^{k_l+1}\nabla f_i(\overline{x}^{k_l+1}) - \frac{1}{n}\sum_{i=1}^n w_i^{k_l}\nabla f_i(x^{k_l}) \end{bmatrix} \in \partial_L F(\overline{w}^{k_l+1}, \overline{x}^{k_l+1}).$$

By the limits $\|\overline{x}^{k+1}(\omega) - x^k(\omega)\|^2 \to 0$ and $\|\overline{w}^{k+1}(\omega) - w^k(\omega)\|^2 \to 0$ as $k \to \infty$ and by continuity, we find that $g^{k_l} \to 0$ as $l \to \infty$. By the definition of the limiting subdifferential (Rockafellar and Wets, 1998, Definition 8.3), it follows that $0 \in \partial_L F(\overline{w}, \overline{x})$. ■

Proof [of Lemma 9] We use the standard descent Lemma, found in (Nesterov, 2004, Lemma 1.2.3), several times throughout the proof.

The result follows by constructing three bounds and adding them together. The first bound: For all $i \in \{1, \ldots, n\}$, we have

$$\begin{aligned} w_i^k f_i(\overline{x}^{k+1}) &\leq w_i^k f_i(x^k) + \langle \overline{x}^{k+1} - x^k, w_i^k \nabla f_i(x^k)\rangle + \frac{|w_i^k| L}{2}\|\overline{x}^{k+1} - x^k\|^2; \\ r_2(\overline{x}^{k+1}) &\leq r_2(x^k) + \langle x^k - \overline{x}^{k+1}, \frac{1}{n}\sum_{i=1}^n w_i^k \nabla f_i(x^k)\rangle - \frac{\eta}{2\gamma}\|\overline{x}^{k+1} - x^k\|^2, \end{aligned}$$

which implies that

$$\begin{aligned}&\frac{1}{n}\sum_{i=1}^{n} w_i^k f_i(\overline{x}^{k+1}) + r_1(\overline{x}^{k+1})\\ &\qquad \le \frac{1}{n}\sum_{i=1}^{n} w_i^k f_i(x^k) + r_1(x^k) + \left[\frac{L}{n}\sum_{i=1}^{n}\frac{|w_i^k|}{2} - \frac{\eta}{2\gamma}\right]\|\overline{x}^{k+1} - x^k\|^2. \qquad (17)\end{aligned}$$

The second bound: For all $i \in \{1, \ldots, n\}$, we have

$$\begin{aligned} w_i^k f_i(\overline{z}^{k+1}) &\le w_i^k f_i(x^k) + \langle \overline{z}^{k+1} - x^k, w_i^k \nabla f_i(x^k)\rangle + \frac{|w_i^k| L}{2}\|\overline{z}^{k+1} - x^k\|^2; \text{ and}\\ w_i^k f_i(x^k) &\le w_i^k f_i(\overline{x}^{k+1}) + \langle x^k - \overline{x}^{k+1}, w_i^k \nabla f_i(x^k)\rangle + \frac{|w_i^k| L}{2}\|\overline{x}^{k+1} - x^k\|^2,\end{aligned}$$

where the second bound follows from the inequality (Nesterov, 2004, Lemma 1.2.3). Adding these bounds together, we obtain

$$w_i^k f_i(\overline{z}^{k+1}) \le w_i^k f_i(\overline{x}^{k+1}) + \langle \overline{z}^{k+1} - \overline{x}^{k+1}, w_i^k \nabla f_i(x^k)\rangle + \frac{|w_i^k| L}{2}\|\overline{z}^{k+1} - x^k\|^2 + \frac{|w_i^k| L}{2}\|\overline{x}^{k+1} - x^k\|^2.$$

Then, by the definition of the proximal operator,

$$r_2(\overline{z}^{k+1}) \le r_2(\overline{x}^{k+1}) + \langle \overline{x}^{k+1} - \overline{z}^{k+1}, v^k\rangle + \frac{1}{2\gamma}\|\overline{x}^{k+1} - x^k\|^2 - \frac{1}{2\gamma}\|\overline{z}^{k+1} - x^k\|^2.$$

Thus,

$$\begin{aligned}&\frac{1}{n}\sum_{i=1}^{n} w_i^k f_i(\overline{z}^{k+1}) + r_1(\overline{z}^{k+1})\\ &\le \frac{1}{n}\sum_{i=1}^{n} w_i^k f_i(\overline{x}^{k+1}) + r_1(\overline{x}^{k+1}) + \left[\frac{L}{n}\sum_{i=1}^{n}\frac{|w_i^k|}{2} - \frac{1}{2\gamma}\right]\|\overline{z}^{k+1} - x^k\|^2\\ &\qquad + \left[\frac{L}{n}\sum_{i=1}^{n}\frac{|w_i^k|}{2} + \frac{1}{2\gamma}\right]\|\overline{x}^{k+1} - x^k\|^2 + \langle \overline{z}^{k+1} - \overline{x}^{k+1}, \frac{1}{n}\sum_{i=1}^{n} w_i^k \nabla f_i(x^k) - v^k\rangle. \qquad (18)\end{aligned}$$

Thus, by adding (17) and (18), we have

$$\begin{aligned}&\frac{1}{n}\sum_{i=1}^{n} w_i^k f_i(\overline{z}^{k+1}) + r_1(\overline{z}^{k+1})\\ &\le \frac{1}{n}\sum_{i=1}^{n} w_i^k f_i(x^k) + r_1(x^k) + \left[\frac{L}{n}\sum_{i=1}^{n}\frac{|w_i^k|}{2} - \frac{1}{2\gamma}\right]\|\overline{z}^{k+1} - x^k\|^2\\ &\qquad + \left[\frac{L}{n}\sum_{i=1}^{n}|w_i^k| - \frac{\eta - 1}{2\gamma}\right]\|\overline{x}^{k+1} - x^k\|^2 + \langle \overline{z}^{k+1} - \overline{x}^{k+1}, \frac{1}{n}\sum_{i=1}^{n} w_i^k \nabla f_i(x^k) - v^k\rangle. \qquad (19)\end{aligned}$$

The third bound: we have

$$\begin{aligned}\frac{1}{n}\sum_{i=1}^n \overline{w}_i^{k+1} f_i(x^k) &= \frac{1}{n}\sum_{i=1}^n w_i^k f_i(x^k) + \langle \overline{w}^{k+1} - w^k, \frac{1}{n}(f_1(x^k),\ldots,f_n(x^k))\rangle;\\ r_1(\overline{w}^{k+1}) &\le r_1(w^k) + \langle w^k - \overline{w}^{k+1}, \frac{1}{n}(f_1(x^k),\ldots,f_n(x^k))\rangle - \frac{1}{2\tau}\|\overline{w}^{k+1} - w^k\|^2,\end{aligned}$$

which implies that

$$\frac{1}{n}\sum_{i=1}^n \overline{w}_i^{k+1} f_i(x^k) + r_1(\overline{w}^{k+1}) \le \frac{1}{n}\sum_{i=1}^n w_i^k f_i(x^k) + r_1(w^k) - \frac{1}{2\tau}\|\overline{w}^{k+1} - w^k\|^2.$$

Therefore, we find that

$$\begin{aligned}&\mathbb{E}_k\left[\frac{1}{n}\sum_{i=1}^n w_i^{k+1} f_i(x^{k+1}) + r_1(w^{k+1}) + r_2(x^{k+1})\right]\\ &\le \frac{q}{n}\sum_{i=1}^n \overline{w}_i^{k+1} f_i(x^k) + r_1(\overline{w}^{k+1}) + r_2(x^k)\\ &+ (1-q)\mathbb{E}_k\left[\frac{1}{n}\sum_{i=1}^n w_i^k f_i(\overline{z}^{k+1}) + r_1(w^k) + r_2(\overline{z}^{k+1})\right]\\ &\le \frac{1}{n}\sum_{i=1}^n w_i^k f_i(x^k) + r_1(w^k) + r_2(x^k) + q'\left[\frac{L}{n}\sum_{i=1}^n \frac{B_i}{2} - \frac{\eta-1}{2\gamma}\right]\|\overline{x}^{k+1} - x^k\|^2\\ &\quad + q'\mathbb{E}_k\left[\left[\frac{L}{n}\sum_{i=1}^n \frac{|w_i^k|}{2} - \frac{1}{2\gamma}\right]\|\overline{z}^{k+1} - x^k\|^2\right] + q'\mathbb{E}_k\left[\langle \overline{z}^{k+1} - \overline{x}^{k+1}, \frac{1}{n}\sum_{i=1}^n w_i^k \nabla f_i(x^k) - v^k\rangle\right]\\ &\quad - \frac{q}{2\tau}\|\overline{w}^{k+1} - w^k\|^2.\end{aligned}$$

■

Proof [of Lemma 10] Noting that

$$\begin{aligned}\xi^k &:= v^k - \frac{1}{n}\sum_{i=1}^n y_i^k\\ \implies \mathbb{E}_k\left[\xi^k\right] &= \mathbb{E}_k\left[\frac{1}{b}\sum_{i\in I_k}(w_i^k\nabla f_i(x^k) - y_i^k)\right] = \frac{1}{n}\sum_{i=1}^n w_i^k \nabla f_i(x^k) - \frac{1}{n}\sum_{i=1}^n y_i^k,\end{aligned}$$

we find that

$$\begin{aligned}
\mathbb{E}_k\left[\left\|\frac{1}{n}\sum_{i=1}^n w_i^k\nabla f_i(x^k) - v^k\right\|^2\right] &= \mathbb{E}_k\left[\left\|\left(\frac{1}{n}\sum_{i=1}^n w_i^k\nabla f_i(x^k) - \frac{1}{n}\sum_{i=1}^n y_i^k\right) - \left(v^k - \frac{1}{n}\sum_{i=1}^n y_i^k\right)\right\|^2\right] \\
&= \mathbb{E}_k\left[\|\xi^k - \mathbb{E}_k\left[\xi^k\right]\|^2\right] \\
&\leq \mathbb{E}_k\left[\|\xi^k\|^2\right] \\
&\leq \mathbb{E}_k\left[\frac{1}{b}\sum_{i\in I_k}\left\|w_i^k\nabla f_i(x^k) - y_i^k\right\|^2\right] \\
&\leq \frac{1}{bn}\sum_{i=1}^n \|w_i^k\nabla f_i(x^k) - y_i^k\|^2.
\end{aligned}$$

where the second to last inequality follows from the bound: for any $z_1, \ldots, z_n \in \mathcal{H}$, we have $\left\|\sum_{i=1}^b z_i\right\|^2 \leq b\sum_{i=1}^b \|z_i\|^2$. ■

Proof [of Lemma 11] Set $q' = (1-q)$. Then

$$\begin{aligned}
&\mathbb{E}_k\left[\left\|w_i^{k+1}\nabla f_i(x^{k+1}) - y_i^{k+1}\right\|^2\right] \\
&= q\left\|\overline{w}_i^{k+1}\nabla f_i(x^k)^T - \overline{w}_i^{k+1}\nabla f_i(x^k)\right\|^2 + q'\rho_i\mathbb{E}_k\left[\left\|w_i^k\nabla f_i(\overline{z}^{k+1}) - w_i^k\nabla f_i(x^k)\right\|^2\right] \\
&\qquad + q'(1-\rho_i)\mathbb{E}_k\left[\left\|w_i^k\nabla f_i(\overline{z}^{k+1}) - y_i^k\right\|^2\right] \\
&\leq (q'\rho_i + q'(1-\rho_i)(1+\beta_i^{-1}))\mathbb{E}_k\left[\left\|w_i^k\nabla f_i(\overline{z}^{k+1}) - w_i^k\nabla f_i(x^k)\right\|^2\right] \\
&\qquad + q'(1-\rho_i)(1+\beta_i)\left\|w_i^k\nabla f_i(x^k) - y_i^k\right\|^2 \\
&\leq q'\left(1 + \frac{1-\rho_i}{\beta_i}\right)\mathbb{E}_k\left[(L_i w_i^k)^2\|\overline{z}^{k+1} - x^k\|^2\right\| + q'(1-\rho_i)(1+\beta_i)\left\|w^k\nabla f_i(x^k) - y_i^k\right\|^2,
\end{aligned}$$

where the first inequality follows from the elementary inequality $\|a+b\|^2 \leq (1+\beta_i^{-1})\|a\|^2 + (1+\beta_i)\|b\|^2$. ■

Proof [of Lemma 12] Let $i \in \{1, \ldots, n\}$ such that $\rho_i \neq 1$. Recall that

$$\beta_i := \sqrt{1-\rho_i}\left(\frac{1}{\sqrt{q'}} - \sqrt{1-\rho_i}\right).$$

Define $\theta_i := 1 - q'(1+\beta_i-\rho_i) = 1 - \sqrt{q'(1-\rho_i)}$ and note that

$$q'(1+\beta_i)(1-\rho_i) \leq q'(1+\beta_i-\rho_i) = \sqrt{q'(1-\rho_i)} = 1-\theta_i.$$

Thus, we have

$$\alpha_i \leq \frac{q'\gamma(1+\epsilon_0)}{2ab}\sum_{t=0}^{\infty}[(1-\theta_i)]^t = \frac{q'\gamma(1+\epsilon_0)}{2ab}\frac{1}{\theta_i} \leq \frac{q'\gamma(1+\epsilon_0)}{2ab\left(1-\sqrt{q'(1-\rho_i)}\right)}.$$

With this bound in hand, we find that

$$\begin{aligned}\alpha_i\left[1+\frac{(1-\rho_i)}{\beta_i}\right] &\leq \frac{q'\gamma(1+\epsilon_0)}{2ab\left(1-\sqrt{q'(1-\rho_i)}\right)}\left[\frac{1}{1-\sqrt{q'(1-\rho_i)}}\right]\\ &= \frac{q'\gamma(1+\epsilon_0)}{2ab\left(1-\sqrt{q'(1-\rho_i)}\right)^2}.\end{aligned}$$

Therefore, because

$$a = \gamma\sqrt{\frac{1}{n}\sum_{i=1}^{n}\frac{q'(1+\epsilon_0)(B_iL_i)^2}{2b\left(1-\sqrt{q'(1-\rho_i)}\right)^2}},$$

we have

$$\begin{aligned}&\frac{1}{n}\sum_{i=1}^{n}\left[q'\alpha_i(Lw_i^k)^2\left(1+\frac{(1-\rho_i)}{\beta_i}\right)+\frac{|w_i^k|L}{2}\right]+\frac{a}{\gamma}\\ &\leq \frac{1}{n}\left[\frac{\gamma}{a}\sum_{i=1}^{n}\frac{q'(1+\epsilon_0)(B_iL)^2}{2b\left(1-\sqrt{q'(1-\rho_i)}\right)^2}+\frac{B_iL}{2}\right]+\frac{a}{\gamma}\\ &= 2L\sqrt{\frac{1}{n}\sum_{i=1}^{n}\frac{q'(1+\epsilon_0)B_i^2}{2b\left(1-\sqrt{q'(1-\rho_i)}\right)^2}}+\frac{L}{n}\sum_{i=1}^{n}\frac{B_i}{2} \leq \frac{1}{2\gamma},\end{aligned}$$

which holds by assumption. ■

A.1 Proof of Corollary 2

Our choice of τ guarantees that $\frac{q\tau}{2} = \frac{q'\gamma}{2\eta}$. Thus, from Part 3 of Theorem 1, it is clear that SMART achieves accuracy ε after $O\left(\frac{2\eta}{q'\gamma\varepsilon}\right)$ iterations. We estimate this ratio below.

Because $D_k \equiv I_k$, we find that $\rho_i = P(i \in D_k) = 1-(1-1/n)^b$. Thus, with

$$a' = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\frac{q'(1+\epsilon_0)(B_iL_i)^2}{2b\left(1-\sqrt{q'(1-\rho_i)}\right)^2}},$$

we have

$$\gamma = \frac{1}{4a' + \frac{L}{n}\sum_{i=1}^{n} B_i},$$
$$\eta = 2\left(1 + 2\gamma\left(a' + \frac{L}{n}\sum_{i=1}^{n} B_i\right)\right),$$

and

$$\begin{aligned}
\frac{\gamma}{2\eta} = \frac{1}{2\eta/\gamma} &= \frac{1}{4(1/\gamma + 2a' + \frac{2L}{n}\sum_{i=1}^{n} B_i)} \\
&= \frac{1}{4(4a' + \frac{L}{n}\sum_{i=1}^{n} B_i + 2a' + \frac{2L}{n}\sum_{i=1}^{n} B_i)} \\
&= \frac{1}{4L\left(\frac{6\sqrt{\frac{1}{n}\sum_{i=1}^{n} q'(1+\epsilon_0)B_i^2}}{\sqrt{2b}\left(1-\sqrt{q'(1-1/n)^b}\right)} + \frac{3}{n}\sum_{i=1}^{n} B_i\right)} \\
&= \frac{\sqrt{2b}\left(1-\sqrt{q'(1-1/n)^b}\right)}{4L\left(6\sqrt{\frac{1}{n}\sum_{i=1}^{n} q'(1+\epsilon_0)B_i^2} + \sqrt{2b}\left(1-\sqrt{q'(1-1/n)^b}\right)\frac{3}{n}\sum_{i=1}^{n} B_i\right)} \\
&= \varOmega\left(\frac{b^{3/2}}{n}\right),
\end{aligned}$$

where we use the bound:

$$1 - \sqrt{q'(1-1/n)^b} = 1 - (1-1/n)^{(b+1)/2} \geq \frac{b+1}{4n}.$$

Therefore, SMART achieves accuracy ε in at most

$$O\left(\frac{2\eta}{q'\gamma\varepsilon}\right) = O\left(\frac{n}{b^{3/2}\varepsilon}\right).$$

iterations.

To initialize properly, SMART requires n gradient evaluations. Then, on average, the w variables will be updated once every n steps, and each of those updates requires n function evaluations, n gradient evaluations, and 1 evaluation of $\mathbf{prox}_{\tau r_1}$. Thus, to reach accuracy ε, SMART requires on average at most $O(n/(b^{3/2}\varepsilon))(1/n)n = O(n/(b^{3/2}\varepsilon))$ function evaluations and $O\left(n/(b^{3/2}\varepsilon)\right)(1/n) = O(1/(b^{3/2}\varepsilon))$ evaluations of $\mathbf{prox}_{\tau r_1}$.

Similarly, the x variables are updated every $1/(1-1/n) = O(1)$ iterations, and each update requires takes b gradient evaluations, and 1 evaluation of $\mathbf{prox}_{\gamma r_1}$. Thus, to reach accuracy ε, SMART requires at most

$$n + O(n/(b^{3/2}\varepsilon))O(1) + O(n/(b^{3/2}\varepsilon))O(b) = O(n + n/(b^{3/2}\varepsilon) + n/(b^{1/2}\varepsilon))$$

gradient evaluations and $O(n/(b^{3/2}\varepsilon))$ evaluations of $\mathbf{prox}_{\gamma r_2}$.

A.2 Proof of Corollary 3

The proof of this Corollary follows the exact same logic as the proof of Corollary 2, up to the equation

$$\begin{aligned}\frac{\gamma}{2\eta} &= \frac{1}{2L\left(\frac{6\sqrt{\frac{1}{n}\sum_{i=1}^{n}(1-1/n)^{b}(1+\epsilon_0)B_i^2}}{\sqrt{2b}\left(1-\sqrt{(1-1/n)^{b}}\right)}+\frac{3}{n}\sum_{i=1}^{n}B_i\right)}\\
&= \frac{\sqrt{2b}\left(1-\sqrt{(1-1/n)^{b}}\right)}{2L\left(6\sqrt{\frac{1}{n}\sum_{i=1}^{n}(1-1/n)^{b}(1+\epsilon_0)B_i^2}+\sqrt{2b}\left(1-\sqrt{(1-1/n)^{b}}\right)\frac{3}{n}\sum_{i=1}^{n}B_i\right)}\\
&= \Omega\left(\frac{b^{3/2}}{n}\right),\end{aligned}$$

where we use the bound:

$$1-\sqrt{(1-1/n)^{b}} = 1-(1-1/n)^{b/2} \geq \frac{b}{4n}.$$

Thus, using the bound $1/q' = 1/(1-1/n)^{b} = O(1/e)$, we find that SMART achieves accuracy ε in at most

$$O\left(\frac{2\eta}{q'\gamma\varepsilon}\right) = O\left(\frac{n}{b^{3/2}\varepsilon}\right).$$

iterations.

To initialize properly, SMART requires n gradient evaluations. Then, on average, the w variables will be updated once every $1/(1-(1-1/n)^{b}) = O(n/b)$ steps, and each of those updates requires n function evaluations, n gradient evaluations, and 1 evaluation of $\mathbf{prox}_{\tau r_1}$. Thus, to reach accuracy ε, SMART requires on average at most $O(n/(b^{3/2}\varepsilon))n(b/n) = O(n/(b^{1/2}\varepsilon))$ function evaluations and $O(n/(b^{3/2}\varepsilon))(b/n) = O(1/(b^{1/2}\varepsilon))$ evaluations of $\mathbf{prox}_{\tau r_1}$.

Similarly, the x variables are updated every $1/(1-1/n)^{b} = O(1)$ iterations, and each update requires b gradient evaluations, and 1 evaluation of $\mathbf{prox}_{\gamma r_2}$. Thus, to reach accuracy ε, SMART requires at most

$$n + O(n/(b^{3/2}\varepsilon))n(b/n) + O(n/(b^{3/2}\varepsilon))b = O(n + n/(b^{1/2}\varepsilon))$$

gradient evaluations and $O(n/(b^{3/2}\varepsilon))$ evaluations of $\mathbf{prox}_{\gamma r_2}$.

Appendix B. Proof of Theorem 4

We use the same notation from the proof of Theorem 1 except that we redefine:

- **The $\delta(\kappa)$ Factor:** $\kappa \in (0, \left[\sqrt{q'(1-\rho_i)}\right]^{-1} - 1)$, we let

$$\delta(\kappa) = \max_i\left\{1-\mu\min\left\{\frac{q'\gamma}{2\eta}, \frac{q\tau}{2}\right\}, (1+\kappa)\sqrt{q'(1-\rho_i)}\right\} \in (0,1),$$

- **The a Factor:**
$$a = \gamma\sqrt{\frac{1}{n}\sum_{i=1}^{n}\frac{q'(1+\epsilon_0)(B_iL_i)^2}{2b\left(\sqrt{\delta(\kappa)}-\sqrt{q'(1-\rho_i)}\right)^2}}.$$
- **The β_i Factor:**
$$\beta_i := \sqrt{1-\rho_i}\left(\frac{\sqrt{\delta(\kappa)}}{\sqrt{q'}}-\sqrt{1-\rho_i}\right).$$
- **The α_i Factor:**
$$\alpha_i := \frac{q'\gamma(1+\epsilon_0)}{2ab\left[\delta(\kappa)-q'(1+\beta_i)(1-\rho_i)\right]}.$$

Note that α_i is well-defined and positive because
$$q'(1+\beta_i)(1-\rho_i) \le q'(1+\beta_i-\rho_i) = \sqrt{q'(1-\rho_i)} \le \delta(\kappa)/(1+\kappa).$$
Then by definition of α_i, we have
$$\alpha_i q'(1+\beta_i)(1-\rho_i) + \frac{q'\gamma(1+\epsilon_0)}{2ab} = \delta(\kappa)\alpha_i. \tag{20}$$

With this choice of α_i and our choice of γ, the following bound holds (we defer the proof for a moment):

Lemma 13 (α_i bound) *The following bound holds: for all $k \in \mathbb{N}$, $\kappa > 0$, we have*
$$\frac{1}{n}\sum_{i=1}^{n}\left[\alpha_i(w_i^kL_i)^2\left(1+\frac{(1-\rho_i)}{\beta_i}\right)+\frac{|w_i^k|L_i}{2}\right] \le \frac{1-2a}{2\gamma}.$$

By nearly the exact same arguments of the proof of Theorem 4 (e.g., recall (16)), we find that
$$\mathbb{E}_k\left[F(w^{k+1},x^{k+1})+\frac{1}{n}\sum_{i=1}^{n}\alpha_i\left\|\nabla c_i(x^{k+1})^Tw_i^{k+1}-y_i^{k+1}\right\|^2\right]$$
$$\le F(w^k,x^k)+\frac{\delta(\kappa)}{n}\sum_{i=1}^{n}\alpha_i\left\|\nabla c_i(x^k)^Tw_i^k-y_i^k\right\|^2-Y_k. \tag{21}$$

where Y_k is defined in (15), and the properties of α_i defined in (20) and Lemma 13 play a key role. From the definition of Y_k and the error bound (5), we find that
$$\begin{aligned}Y_k &\ge \frac{q'\gamma}{2\eta}\left\|\frac{\eta}{\gamma}\left(\overline{x}^{k+1}-x^k\right)\right\|^2+\frac{q\tau}{2}\left\|\frac{1}{\tau}\left(\overline{w}^{k+1}-w^k\right)\right\|^2\\ &\ge \min\left\{\frac{q'\gamma}{2\eta},\frac{q\tau}{2}\right\}\mu\left[F(w^k,x^k)-F(w^*,x^*)\right]\\ &\ge (1-\delta(\kappa))\mu\left[F(w^k,x^k)-F(w^*,x^*)\right].\end{aligned}$$

Thus, by plugging this bound into (21), we have

$$\mathbb{E}_k\left[F(w^{k+1},x^{k+1})-F(w^*,x^*)+\frac{1}{n}\sum_{i=1}^n\alpha_i\left\|\nabla c_i(x^{k+1})^Tw_i^{k+1}-y_i^{k+1}\right\|^2\right]$$
$$\leq\delta(\kappa)\left[F(w^k,x^k)-F(w^*,x^*)+\frac{1}{n}\sum_{i=1}^n\alpha_i\left\|\nabla c_i(x^k)^Tw_i^k-y_i^k\right\|^2\right].$$

To complete the proof, we use the law of total expectation to unfold the contraction: for all $k\in\mathbb{N}$, we have

$$\begin{aligned}\mathbb{E}\left[F(w^{k+1},x^{k+1})-F(w^*,x^*)\right]&\leq\mathbb{E}\left[X_{k+1}-F(w^*,x^*)\right]\\&\leq\delta(\kappa)\mathbb{E}\left[X_k-F(w^*,x^*)\right]\\&\leq\delta(\kappa)^{k+1}\left[X_0-F(w^*,x^*)\right]=\delta(\kappa)^{k+1}\left[F(w^0,x^0)-F(w^*,x^*)\right].\end{aligned}$$

Take the limit as $\kappa\to 0$ to get the result.
Proof [of Lemma 13] Let $i\in\{1,\ldots,n\}$ such that $\rho_i\neq 1$. Recall that

$$\beta_i:=\sqrt{1-\rho_i}\left(\frac{\sqrt{\delta(\kappa)}}{\sqrt{q'}}-\sqrt{1-\rho_i}\right).$$

Define $\theta_i:=1-(1/\delta(\kappa))q'(1+\beta_i-\rho_i)=1-(1/\sqrt{\delta(\kappa)})\sqrt{q'(1-\rho_i)}$ and note that

$$\frac{q'(1+\beta_i)(1-\rho_i)}{\delta(\kappa)}\leq\frac{q'(1+\beta_i-\rho_i)}{\delta(\kappa)}=\frac{\sqrt{q'(1-\rho_i)}}{\sqrt{\delta(\kappa)}}=1-\theta_i.$$

Thus, we have

$$\alpha_i\leq\frac{q'\gamma(1+\epsilon_0)}{2\delta(\kappa)ab}\sum_{t=0}^\infty[(1-\theta_i)]^t=\frac{q'\gamma(1+\epsilon_0)}{2\delta(\kappa)ab}\frac{1}{\theta_i}\leq\frac{q'\gamma(1+\epsilon_0)}{2ab\sqrt{\delta(\kappa)}\left(\sqrt{\delta(\kappa)}-\sqrt{q'(1-\rho_i)}\right)}.$$

With this bound in hand, we find that

$$\begin{aligned}\alpha_i\left[1+\frac{(1-\rho_i)}{\beta_i}\right]&\leq\frac{q'\gamma(1+\epsilon_0)}{2ab\sqrt{\delta(\kappa)}\left(\sqrt{\delta(\kappa)}-\sqrt{q'(1-\rho_i)}\right)}\left[\frac{\sqrt{\delta(\kappa)}}{\sqrt{\delta(\kappa)}-\sqrt{q'(1-\rho_i)}}\right]\\&=\frac{q'\gamma(1+\epsilon_0)}{2ab\left(\sqrt{\delta(\kappa)}-\sqrt{q'(1-\rho_i)}\right)^2}.\end{aligned}$$

Therefore, because

$$a=\gamma\sqrt{\frac{1}{n}\sum_{i=1}^n\frac{q'(1+\epsilon_0)(B_iL_i)^2}{2b\left(\sqrt{\delta(\kappa)}-\sqrt{q'(1-\rho_i)}\right)^2}},$$

we have

$$\begin{aligned}
&\frac{1}{n}\sum_{i=1}^{n}\left[q'\alpha_i(L_i w_i^k)^2\left(1+\frac{(1-\rho_i)}{\beta_i}\right)+\frac{|w_i^k|L_i}{2}\right]+\frac{a}{\gamma}\\
&\leq \frac{1}{n}\left[\frac{\gamma}{a}\sum_{i=1}^{n}\frac{q'(1+\epsilon_0)(B_iL_i)^2}{2b\left(\sqrt{\delta(\kappa)}-\sqrt{q'(1-\rho_i)}\right)^2}+\frac{B_iL_i}{2}\right]+\frac{a}{\gamma}\\
&= 2\sqrt{\frac{1}{n}\sum_{i=1}^{n}\frac{q'(1+\epsilon_0)(B_iL_i)^2}{2b\left(\sqrt{\delta(\kappa)}-\sqrt{q'(1-\rho_i)}\right)^2}}+\frac{1}{n}\sum_{i=1}^{n}\frac{B_iL_i}{2}\\
&\leq 2\sqrt{\frac{1}{n}\sum_{i=1}^{n}\frac{q'(1+\epsilon_0)(B_iL_i)^2}{2b\sqrt{q'(1-\rho_i)}\left(1-(q'(1-\rho_i))^{1/4}\right)^2}}+\frac{1}{n}\sum_{i=1}^{n}\frac{B_iL_i}{2}\leq\frac{1}{2\gamma},
\end{aligned}$$

where the second to last line follows because

$$\delta(\kappa)\geq(1+\kappa)\sqrt{q'(1-\rho_i)}\geq\sqrt{q'(1-\rho_i)},$$

and the last line holds by assumption.

■

B.1 Proof of Corollary 5

Our choice of τ guarantees that $\frac{q\tau}{2}=\frac{q'\gamma}{2\eta}$. Thus, from Theorem 4, it is clear that SMART achieves accuracy ε after

$$O\left(\log(1/\epsilon)/\log(1/\delta)\right))$$

iterations. We estimate this ratio below.

Because $D_k\equiv I_k$, we find that $\rho_i=P(i\in D_k)=1-(1-1/n)^b$. Thus, with

$$a'=\sqrt{\frac{1}{n}\sum_{i=1}^{n}\frac{q'(1+\epsilon_0)(B_iL_i)^2}{2b\sqrt{q'(1-\rho_i)}\left(1-(q'(1-\rho_i))^{1/4}\right)^2}},$$

we have

$$\begin{aligned}
\frac{\gamma}{2\eta} &= \frac{1}{2(1/\gamma + 2a' + 2L\sum_{i=1}^n B_i)} \\
&= \frac{1}{2(4a' + \frac{L}{n}\sum_{i=1}^n B_i + 2a' + \frac{2L}{n}\sum_{i=1}^n B_i)} \\
&= \frac{1}{2L\left(\frac{6\sqrt{\frac{1}{n}\sum_{i=1}^n q'(1+\epsilon_0)B_i^2}}{\sqrt{2b}(1-1/n)^{(b+1)/4}\left(1-(1-1/n)^{(b+1)/4}\right)} + \frac{3}{n}\sum_{i=1}^n B_i\right)} \\
&= \frac{\sqrt{2b}(1-1/n)^{(b+1)/4}\left(1-(1-1/n)^{(b+1)/4}\right)}{2L\left(6\sqrt{\frac{1}{n}\sum_{i=1}^n q'(1+\epsilon_0)B_i^2} + \sqrt{2b}(1-1/n)^{(b+1)/4}\left(1-(1-1/n)^{(b+1)/4}\right)\frac{3}{n}\sum_{i=1}^n B_i\right)} \\
&= \Omega\left(\frac{b^{3/2}}{Ln}\right),
\end{aligned}$$

where we use the bounds:

$$1-(1-1/n)^{(b+1)/2} \geq \frac{b+1}{8n}; \text{ and}$$
$$(1-1/n)^{(b+1)/4} = \Omega(1/e).$$

Thus,

$$\frac{1}{\log\left(\frac{1}{1-\frac{\mu q'\gamma}{2L\eta}}\right)} \leq \frac{2\eta L}{\mu q'\gamma} = O\left(\kappa\frac{n}{b^{3/2}}\right).$$

Similarly

$$\frac{1}{\log\left(\frac{1}{\sqrt{q'(1-\rho_i)}}\right)} = \frac{1}{\log\left(\frac{1}{(1-1/n)^{(b+1)/2}}\right)} \leq \frac{1}{1-(1-1/n)^{(b+1)/2}} = O\left(\frac{n}{b^{3/2}}\right)$$

Therefore, SMART achieves accuracy ε in at most

$$\frac{\log(1/\varepsilon)}{\log(1/\delta)} = O\left(\kappa\frac{n}{b^{3/2}}\log(1/\epsilon)\right)$$

iterations.

To initialize properly, SMART requires n gradient evaluations. Then, on average, the w variables will be updated once every n steps, and each of those updates requires n function evaluations, n gradient evaluations, and evaluation of $\mathbf{prox}_{\tau r_1}$. Thus, to reach accuracy ε, SMART requires on average at most $O(\kappa(n/b^{3/2})\log(1/\varepsilon))$ function evaluations and $O(\kappa(1/b^{3/2})\log(1/\varepsilon))$ evaluations of $\mathbf{prox}_{\tau r_1}$. Similarly, the x variables are updated every $1/(1-1/n) = O(1)$ iterations, and each update requires takes b gradient evaluations, and 1 evaluation of $\mathbf{prox}_{\gamma r_1}$. Thus, to reach accuracy ε, SMART requires at most $O(n + \kappa(n/b^{3/2} + n/b^{1/2})\log(1/\varepsilon))$ gradient evaluations and $O\left(\kappa(n/b^{3/2})\log(1/\varepsilon)\right)$ evaluations of $\mathbf{prox}_{\gamma r_2}$.