Accelerating Empirical Risk Minimization: Unified Approaches with Proximal Point and Stochastic Optimization Methods

Chris Junchi Li$^\diamond$

Department of Electrical Engineering and Computer Sciences$^\diamond$
University of California, Berkeley

October 6, 2024

Abstract

In this paper, we propose a novel framework for solving empirical risk minimization (ERM) problems by introducing un-regularized approaches to proximal point algorithms. Our approach leverages stochastic optimization methods, providing both theoretical guarantees and practical improvements over existing techniques. By systematically applying approximate variants of the proximal point algorithm (PPA) combined with dual ascent methods, we achieve accelerated convergence rates for strongly convex and smooth objectives, even in high-dimensional settings. We highlight the flexibility of the framework by showcasing its application to both primal and dual formulations, leading to superior performance in empirical risk minimization tasks.

Keywords: Empirical Risk Minimization, Proximal Point Algorithm, Stochastic Optimization, Dual Ascent, Acceleration Techniques.

1 Introduction

Empirical Risk Minimization (ERM) lies at the heart of many machine learning tasks, where the goal is to find a predictor or regressor that minimizes the cumulative loss over a given dataset. This problem has gained significant attention due to its central role in supervised learning, ranging from logistic regression to linear least-squares models. Traditionally, optimization methods that solve ERM problems rely heavily on gradient-based techniques, but these methods often suffer from poor convergence rates when the condition number of the problem is large, which is common in high-dimensional data.

In recent years, randomized and stochastic optimization algorithms have emerged as promising alternatives to deterministic first-order methods. Among these, stochastic variance reduction techniques and dual ascent methods have demonstrated faster convergence for well-conditioned problems. However, their performance degrades when dealing with poorly conditioned data, as they often require explicit regularization to achieve acceleration.

In this paper, we address these challenges by introducing a new un-regularized approach to proximal point algorithms. By incorporating approximate minimization strategies into the proximal point framework, we develop a suite of stochastic algorithms that achieve accelerated convergence without the need for explicit regularization. Our framework offers a flexible and efficient solution for a wide variety of convex optimization problems, particularly those encountered in large-scale machine learning.

Formulations We focus in part on the problem of empirical risk minimization of linear predictors: given a set of n data points $a_i, \ldots, a_n \in \mathbb{R}^d$ and convex loss functions $\phi_i : \mathbb{R} \to \mathbb{R}$ for $i = 1, \ldots, n$, solve

$$\min_{x \in \mathbb{R}^n} F(x), \qquad \text{where} \qquad F(x) \stackrel{\text{def}}{=} \sum_{i=1}^{n} \phi_i(a_i^{\mathsf{T}} x). \tag{1}$$

This problem underlies supervised learning (*e.g.* the training of logistic regressors when $\phi_i(z) = \log(1 + e^{-zb_i})$, or their regularized form when $\phi_i(z) = \log(1 + e^{-zb_i}) + \frac{\gamma}{2n}\|x\|_2^2$ for a scalar $\gamma > 0$) and captures the widely-studied problem of linear least-squares regression when $\phi_i(z) = \frac{1}{2}(z - b_i)^2$.

Over the past five years, problems such as (1) have received increased attention, with a recent burst of activity in the design of fast *randomized* algorithms. Iterative methods that randomly sample the ϕ_i have been shown to outperform standard first-order methods under mild assumptions [Bottou and Bousquet(2008), Johnson and Zhang(2013), Xiao and Zhang(2014), Defazio et al.(2014), Shalev-Shwartz and Zhang(2014)].

Despite the breadth of these recent results, their running time guarantees when solving the ERM problem (1) are sub-optimal in terms of their dependence on a natural notion of the problem's *condition number* (See Section 1.1). This dependence can, however, significantly impact their guarantees on running time. High-dimensional problems encountered in practice are often poorly conditioned. In large-scale machine learning applications, the condition number of the ERM problem (1) captures notions of data complexity arising from variable correlation in high dimensions and is hence prone to be very large.

More specifically, among the recent randomized algorithms, each one either:

1. Solves the ERM problem (1), under an assumption of strong convexity, with convergence that depends linearly on the problem's condition number [Johnson and Zhang(2013), Defazio et al.(2014)].

2. Solves only an explicitly *regularized* ERM problem, $\min_x\{F(x) + \lambda r(x)\}$ where the regularizer r is a known 1-strongly convex function and λ must be strictly positive, even when F is itself strongly convex. One such result is due to [Shalev-Shwartz and Zhang(2014)] and is the first to achieve *acceleration* for this problem, *i.e.* dependence only on the square root of the regularized problem's condition number, which scales inversely with λ. Hence, taking small λ to solve the ERM problem (where $\lambda = 0$ in effect) is not a viable option.

In this paper we show how to bridge this gap via black-box reductions. Namely, we develop algorithms to solve the ERM problem (1) – under a standard assumption of strong convexity – through repeated, approximate minimizations of the regularized ERM problem $\min_x\{F(x) + \lambda r(x)\}$ for fairly large λ. Instantiating our framework with known randomized algorithms that solve the regularized ERM problem, we achieve accelerated running time guarantees for solving the original ERM problem.

The key to our reductions are approximate variants of the classical *proximal point algorithm* (PPA) [Rockafellar(1976), Parikh and Boyd(2014)]. We show how both PPA and the inner minimization procedure can then be accelerated and our analysis gives precise approximation requirements for either option. Furthermore, we show further practical improvements when the inner minimizer operates by a dual ascent method. In total, this provides at least three different algorithms for achieving an improved accelerated running time for solving the ERM problem (1) under the standard assumption of strongly convex F and smooth ϕ_i. (Table 1 summarizes our improvements in comparison to existing minimization procedures.)

Perhaps the strongest and most general theoretical reduction we provide in this paper is encompassed by the following theorem which we prove in Section 3.

Theorem 1.1 (Accelerated Approximate Proximal Point Algorithm)**.** *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a μ-strongly convex function and suppose that, for all $x_0 \in \mathbb{R}^n$, $c > 0$, $\lambda > 0$, we can compute a point x_c (possibly random) such that*

$$\mathbb{E}f(x_c) - \min_x \left\{ f(x) + \frac{\lambda}{2}\|x - x_0\|_2^2 \right\} \leq \frac{1}{c}\left[f(x_0) - \min_x \left\{ f(x) + \frac{\lambda}{2}\|x - x_0\|_2^2 \right\} \right]$$

in time $\mathcal{T}_c$. Then given any x_0, $c > 0$, $\lambda \geq 2\mu$, we can compute x_1 such that

$$\mathbb{E}f(x_1) - \min_x f(x) \leq \frac{1}{c}\left[f(x_0) - \min_x f(x) \right]$$

in time $O\left(\mathcal{T}_{4(\frac{2\lambda+\mu}{\mu})^{3/2}} \sqrt{\lceil \lambda/\mu \rceil} \log c \right)$.

This theorem essentially states that we can use a linearly convergent algorithm for minimizing $f(x) + \lambda\|x - x_0\|_2^2$ in order to minimize f, while incurring a multiplicative overhead of only $O(\sqrt{\lceil \lambda/\mu \rceil}\,\mathrm{polylog}(\lambda/\mu))$. Applying this theorem to previous state-of-the-art algorithms improves both the running time for solving (1), as well as the following more general ERM problem:

$$\min_{x \in \mathbb{R}^d} \sum_{i=1}^{n} \psi_i(x), \qquad \text{where} \qquad \psi_i : \mathbb{R}^d \to \mathbb{R}. \tag{2}$$

Problem (2) is fundamental in the theory of convex optimization and covers ERM problems for multiclass and structured prediction.

There are a variety of additional extensions to the ERM problem to which some of our analysis easily applies. For instance, we could work in more general normed spaces, allow non-uniform smoothness of the ϕ, add an explicit regularizer, etc. However, to simplify exposition and comparison to related work, we focus on (1) and make clear the extensions to (2) in Section 3. These cases capture the core of the arguments presented and illustrate the generality of this approach.

Several of the algorithmic tools and analysis techniques in this paper are similar in principle to (and sometimes appear indirectly in) work scattered throughout the machine learning and optimization literature – from classical treatments of error-tolerant PPA [Rockafellar(1976), Guler(1992)] to the effective proximal term used by Accelerated Proximal SDCA [Shalev-Shwartz and Zhang(2014)] in enabling its acceleration.

By analyzing these as separate tools, and by bookkeeping the error requirements that they impose, we are able to assemble them into algorithms with improved guarantees. We believe that the presentation of Accelerated APPA (Algorithm 2) arising from this view simplifies, and clarifies in terms of broader convex optimization theory, the "outer loop" steps employed by Accelerated Proximal SDCA. More generally, we hope that disentangling the relevant algorithmic components into this general reduction framework will lead to further applications both in theory and in practice.

1.1 Formal setup

We consider the ERM problem (1) in the following common setting:

Assumption 1.2 (Regularity)**.** *Each loss function ϕ_i is L-smooth, i.e. for all $x, y \in \mathbb{R}$,*

$$\phi(y) \leq \phi(x) + \phi'(x)(y - x) + \frac{L}{2}(y - x)^2,$$

and the sum F is μ-strongly convex, i.e. for all $x, y \in \mathbb{R}^d$,

$$F(x) \geq F(x) + \nabla F(x)^\top (y-x) + \frac{\mu}{2}\|y-x\|_2^2.$$

We let $R \stackrel{\text{def}}{=} \max_i \|a_i\|_2$ and let $A \in \mathbb{R}^{n \times d}$ be the matrix whose i'th row is $a_i^\top$. We refer to

$$\kappa \stackrel{\text{def}}{=} \lceil LR^2/\mu \rceil$$

as the *condition number* of (1).

Although many algorithms are designed for special cases of the ERM objective F where there is some known, exploitable structure to the problem, our aim is to study the most general case subject to Assumption 1.2. To standardize the comparison among algorithms, we consider the following generic model of interaction with F:

Assumption 1.3 (Computational model)**.** *For any $i \in [n]$ and $x \in \mathbb{R}^d$, we consider two primitive operations:*

- *For $b \in \mathbb{R}$, compute the gradient of $x \mapsto \phi_i(a_i^\top x - b)$.*
- *For $b \in \mathbb{R}$, $c \in \mathbb{R}^d$, minimize $\phi_i(a_i^\top x) + b\|x-c\|_2^2$.*

We refer to these operations, as well as to the evaluation of $\phi_i(a_i^\top x)$, as single accesses *to ϕ_i, and assume that these operations can be performed in $O(d)$ time.*

Notation Denote $[n] \stackrel{\text{def}}{=} \{1, \ldots, n\}$. Denote the optimal value of a convex function by $f^{\text{opt}} = \min_x f(x)$, and, when f is clear from context, let x^{opt} denote a minimizer. A point x' is an *ϵ-approximate minimizer* of f if $f(x') - f^{\text{opt}} \leq \epsilon$. The Fenchel dual of a convex function $f : \mathbb{R}^k \to \mathbb{R}$ is $f^* : \mathbb{R}^k \to \mathbb{R}$ defined by $f^*(y) = \sup_{x \in \mathbb{R}^k}\{\langle y, x\rangle - f(x)\}$. We use $\widetilde{O}(\cdot)$ to hide factors polylogarithmic in n, L, μ, λ, and R, *i.e.* $\widetilde{O}(f) = O(f \operatorname{polylog}(n, L, \mu, \lambda, R))$.

Regularization and duality Throughout the paper we let $F : \mathbb{R}^d \to \mathbb{R}$ denote a μ-strongly convex function. For certain results presented, F must in particular be the ERM problem (1), while other statements hold more generally. We make it clear on a case-by-case basis when F must have the ERM structure as in (1).

Beginning in Section 1.3 and throughout the remainder of the paper, we frequently consider the function $f_{s,\lambda}(x)$, defined for all $x, s \in \mathbb{R}^d$ and $\lambda > 0$ by

$$f_{s,\lambda}(x) \stackrel{\text{def}}{=} F(x) + \tfrac{\lambda}{2}\|x-s\|_2^2 \tag{3}$$

In such context, we let $x_{s,\lambda}^{\text{opt}} \stackrel{\text{def}}{=} \operatorname{argmin}_x f_{s,\lambda}(x)$ and we call

$$\kappa_\lambda \stackrel{\text{def}}{=} \lceil LR^2/\lambda \rceil$$

the *regularized condition number.*

When F is indeed the ERM objective (1), certain algorithms for minimizing $f_{s,\lambda}$ operate in the *regularized ERM dual.* Namely, they proceed by decreasing the negative dual objective $g_{s,\lambda} : \mathbb{R}^n \to \mathbb{R}$, given by

$$g_{s,\lambda}(y) \stackrel{\text{def}}{=} G(y) + \frac{1}{2\lambda}\|A^\top y\|_2^2 - s^\top A^\top y, \tag{4}$$

Empirical risk minimization		
Algorithm	Running time	Problem
GD	$dn^2\kappa\log(\epsilon_0/\epsilon)$	F
Accel. GD	$dn^{3/2}\sqrt{\kappa}\log(\epsilon_0/\epsilon)$	F
SAG, SVRG	$dn\kappa\log(\epsilon_0/\epsilon)$	F
SDCA	$dn\kappa_\lambda\log(\epsilon_0/\epsilon)$	$F+\lambda r$
AP-SDCA	$dn\sqrt{\kappa_\lambda}\log(\epsilon_0/\epsilon)$	$F+\lambda r$
APCG	$dn\sqrt{\kappa_\lambda}\log(\epsilon_0/\epsilon)$	$F+\lambda r$
This work	$dn\sqrt{\kappa}\log(\epsilon_0/\epsilon)$	F

Linear least-squares regression		
Algorithm	Running time	Problem
Naive mult.	nd^2	$\|Ax-b\|_2^2$
Fast mult.	$nd^{\omega-1}$	$\|Ax-b\|_2^2$
Row sampling	$(nd+d^\omega)\log(\epsilon_0/\epsilon)$	$\|Ax-b\|_2^2$
OSNAP	$(nd+d^\omega)\log(\epsilon_0/\epsilon)$	$\|Ax-b\|_2^2$
R. Kaczmarz	$dn\kappa\log(\epsilon_0/\epsilon)$	$Ax=b$
Acc. coord.	$dn\sqrt{\kappa}\log(\epsilon_0/\epsilon)$	$Ax=b$
This work	$dn\sqrt{\kappa}\log(\epsilon_0/\epsilon)$	$\|Ax-b\|_2^2$

Table 1. Theoretical performance comparison on ERM and linear regression. Running times hold in expectation for randomized algorithms. In the "problem" column for ERM, F marks algorithms that can optimize the ERM objective (1), while $F+\lambda r$ marks those that only solve the explicitly regularized problem. For linear regression, $Ax=b$ marks algorithms that only solve consistent linear systems, whereas $\|Ax-b\|_2^2$ marks those that more generally minimize the squared loss. The constant ω denotes the exponent of the matrix multiplication running time (currently below 2.373 [Williams(2012)]). See Section 1.2 for more detail on these algorithms and their running times.

where $G(y) \overset{\text{def}}{=} \sum_{i=1}^n \phi_i^*(y_i)$. Similar to the above, we let $y_{s,\lambda}^{\text{opt}} \overset{\text{def}}{=} \operatorname{argmin}_y g_{s,\lambda}(y)$.

To make corresponding primal progress, dual-based algorithms make use of the *dual-to-primal* mapping, given by

$$\widehat{x}_{s,\lambda}(y) \overset{\text{def}}{=} s - \tfrac{1}{\lambda}A^\mathsf{T}y, \tag{5}$$

and the *primal-to-dual* mapping, given entrywise by

$$[\widehat{y}(x)]_i \overset{\text{def}}{=} \left[\frac{\partial\phi_i(z)}{\partial z}\right]\Big|_{z=a_i^\mathsf{T}x} \tag{6}$$

for $i=1,\ldots,n$. (See Appendix B for a derivation of these facts and further properties of the dual.)

1.2 Running times and related work

In Table 1 we compare our results with the running time of both classical and recent algorithms for solving the ERM problem (1) and linear least-squares regression. Here we briefly explain these running times and related work.

Empirical risk minimization In the context of the ERM problem, GD refers to canonical gradient descent on F, Accel. GD is Nesterov's accelerated gradient decent [Nesterov(1983), Nesterov(2004)], SVRG is the stochastic variance-reduced gradient of [Johnson and Zhang(2013)], SAG is the stochastic average gradient of [Roux et al.(2012)] and [Defazio et al.(2014)], SDCA is the stochastic dual coordinate ascent of [Shalev-Shwartz and Zhang(2013)], AP-SDCA is the Accelerated Proximal SDCA of [Shalev-Shwartz and Zhang(2014)] and APCG is the accelerated coordinate algorithm of [Lin et al.(2014)]. The latter three algorithms are more restrictive in that they only solve the explicitly regularized problem $F+\lambda r$, even if F is itself strongly convex (such algorithms run in time inversely proportional to λ).

The running times of the algorithms are presented based on the setting considered in this paper, *i.e.* under Assumptions 1.2 and 1.3. Many of the algorithms can be applied in more general settings (*e.g.* even if the function F is not strongly convex) and have different convergence guarantees in those cases. The running times are characterized by four parameters: d is the data dimension, n is the number of samples, $\kappa = \lceil LR^2/\mu \rceil$ is the condition number (for $F + \lambda r$ minimizers, the condition number $\kappa_\lambda = \lceil LR^2/\lambda \rceil$ is used) and ϵ_0/ϵ is the ratio between the initial and desired accuracy. Running times are stated per $\widetilde{O}$-notation; factors that depend polylogarithmically on n, κ, and κ_λ are ignored.

Linear least-squares regression For the linear least-squares regression problem, there is greater variety in the algorithms that apply. For comparison, Table 1 includes Moore-Penrose pseudoinversion – computed via naive matrix multiplication and inversion routines, as well as by their asymptotically fastest counterparts – in order to compute a closed-form solution via the standard normal equations. The table also lists algorithms based on the randomized Kaczmarz method [Strohmer and Vershynin(2009), Needell et al.(2014)] and their accelerated variant [Lee and Sidford(2013)], as well as algorithms based on subspace embedding (OSNAP) or row sampling [Nelson and Nguyen(2013), Li et al.(2013), Cohen et al.(2015)]. Some Kaczmarz-based methods only apply to the more restrictive problem of solving a consistent system (finding x satisfying $Ax = b$) rather than minimize the squared loss $\|Ax - b\|_2^2$. The running times depend on the same four parameters $n, d, \kappa, \epsilon_0/\epsilon$ as before, except for computing the closed-form pseudoinverse, which for simplicity we consider "exact," independent of initial and target errors ϵ_0/ϵ.

Approximate proximal point The key to our improved running times is a suite of approximate proximal point algorithms that we propose and analyze. We remark that notions of error-tolerance in the typical proximal point algorithm – for both its plain and accelerated variants – have been defined and studied in prior work [Rockafellar(1976), Guler(1992)]. However, these mainly consider the cumulative *absolute* error of iterates produced by inner minimizers, assuming that such a sequence is somehow produced. Since essentially any procedure of interest begins at some initial point – and has runtime that depends on the *relative* error ratio between its start and end – such a view does not yield fully concrete algorithms, nor does it yield end-to-end runtime upper bounds such as those presented in this paper.

Additional related work There is an immense body of literature on proximal point methods and alternating direction method of multipliers (ADMM) that are relevant to the approach in this paper; see [Boyd et al.(2011), Parikh and Boyd(2014)] for modern surveys. We also note that the independent work of [Lin et al.(2015)] contains results similar to some of those in this paper.

1.3 Main results

All formal results in this paper are obtained through a framework that we develop for iteratively applying and accelerating various minimization algorithms. When instantiated with recently-developed fast minimizers we obtain, under Assumptions 1.2 and 1.3, algorithms guaranteed to solve the ERM problem in time $\widetilde{O}(nd\sqrt{\kappa}\log(1/\epsilon))$.

Our framework stems from a critical insight of the classical *proximal point algorithm (PPA)* or *proximal iteration*: to minimize F (or more generally, any convex function) it suffices to iteratively minimize

$$f_{s,\lambda}(x) \overset{\text{def}}{=} F(x) + \tfrac{\lambda}{2}\|x - s\|_2^2$$

for $\lambda > 0$ and proper choice of *center* $s \in \mathbb{R}^d$. PPA iteratively applies the update

$$x^{(t+1)} \leftarrow \operatorname*{argmin}_x f_{x^{(t)},\lambda}(x)$$

and converges to the minimizer of F. The minimization in the update is known as the *proximal operator* [Parikh and Boyd(2014)], and we refer to it in the sequel as the *inner* minimization problem.

In this paper we provide three distinct *approximate* proximal point algorithms, *i.e.* algorithms that do not require full inner minimization. Each enables the use of existing fast algorithm as its inner minimizer, in turn yielding several ways to obtain our improved ERM running time:

- In Section 2 we develop a basic approximate proximal point algorithm (APPA). The algorithm is essentially PPA with a relaxed requirement of inner minimization by only a *fixed* multiplicative constant in each iteration. Instantiating this algorithm with an accelerated, regularized ERM solver – such as APCG [Lin et al.(2014)] – as its inner minimizer yields the improved accelerated running time for the ERM problem (1).

- In Section 3 we develop Accelerated APPA. Instantiating this algorithm with SVRG [Johnson and Zhang(2013 as its inner minimizer yields the improved accelerated running time for both the ERM problem (1) as well as the general ERM problem (2).

- In Section 4 we develop Dual APPA: an algorithm whose approximate inner minimizers operate on the dual $f_{s,\lambda}$, with warm starts between iterations. Dual APPA enables several inner minimizers that are a priori incompatible with APPA. Instantiating this algorithm with an accelerate, regularized ERM solver – such as APCG [Lin et al.(2014)] – as its inner minimizer yields the improved accelerated running time for the ERM problem (1).

Each of the three algorithms exhibits a slight advantage over the others in different regimes. APPA has by far the simplest and most straightforward analysis, and applies directly to any μ-strongly convex function F (not only F given by (1)). Accelerated APPA is more complicated, but in many regimes is a more efficient reduction than APPA; it too applies to any μ-strongly convex function F and in turn proves Theorem 1.1.

Our third algorithm, Dual APPA, is the least general in terms of the assumptions on which it relies. It is the only reduction we develop that requires the ERM structure of F. However, this algorithm is a natural choice in conjunction with inner minimizers that operate on a popular dual objective.

1.4 Paper organization

The remainder of this paper is organized as follows. In Section 2, Section 3, and Section 4 we state and analyze the approximate proximal point algorithms described above. In Appendix A we prove general technical lemmas used throughout the paper and in Appendix B we provide a derivation of regularized ERM duality and related technical lemmas.

2 Approximate proximal point algorithm (APPA)

In this section we describe our approximate proximal point algorithm (APPA). This algorithm is perhaps the simplest, both in its description and in its analysis, in comparison to the others

Algorithm 1 Approximate PPA (APPA)

```
Input: $x^{(0)} \in \mathbb{R}^d$, $\lambda > 0$
Input: primal $(\frac{2(\lambda+\mu)}{\mu}, \lambda)$-oracle $\mathcal{P}$
for $t = 1, \ldots, T$ do
    $x^{(t)} \leftarrow \mathcal{P}(x^{(t-1)})$
end for
OUTPUT: $x^{(T)}$
```

described in this paper. This section also introduces technical machinery that is used throughout the sequel.

We first present our formal abstraction of inner minimizers (Section 2.1), then we present our algorithm (Section 2.2), and finally we step through its analysis (Section 2.3).

2.1 Approximate primal oracles

To design APPA, we first quantify the error that can be tolerated of an inner minimizer, while accounting for the computational cost of ensuring such error. The abstraction we use is the following notion of inner approximation:

$:=$ An algorithm $\mathcal{P}$ is a *primal* (c, λ)*-oracle* if, given $x \in \mathbb{R}^d$, it outputs $\mathcal{P}(x)$ that is a $([f_{x,\lambda}(x) - f^{\text{opt}}_{x,\lambda}]/c)$-approximate minimizer of $f_{x,\lambda}$ in time $\mathcal{T}_{\mathcal{P}}$.[1]

In other words, a primal oracle is an algorithm initialized at x that reduces the error of $f_{x,\lambda}$ by a $1/c$ fraction, in time that depends on λ, and c, and regularity properties of F. Typical iterative first-order algorithms, such as those in Table 1, yield primal (c, λ)-oracles with runtimes $\mathcal{T}_{\mathcal{P}}$ that scale inversely in λ or $\sqrt{\lambda}$, and logarithmically in c. For instance:

Theorem 2.1 (SVRG as a primal oracle)**.** *SVRG [Johnson and Zhang(2013)] is a primal* (c, λ)*-oracle with runtime complexity* $\mathcal{T}_{\mathcal{P}} = O(nd \min\{\kappa, \kappa_\lambda\} \log c)$ *for both the ERM problem* (1) *and the general ERM problem* (2)*.*

Theorem 2.2 (APCG as an accelerated primal oracle)**.** *Using APCG [Lin et al.(2014)] we can obtain a primal* (c, λ)*-oracle with runtime complexity* $\mathcal{T}_{\mathcal{P}} = \widetilde{O}(nd\sqrt{\kappa_\lambda} \log c)$ *for the ERM problem* (1)*.*[2]

Proof. Corollary B.3 implies that, given a primal point x, we can obtain, in $O(nd)$ time, a corresponding dual point y such that the duality gap $f_{x,\lambda}(x) + g_{x,\lambda}(y)$ (and thus the dual error) is at most $O(\text{poly}(\kappa_\lambda))$ times the primal error. Lemma B.1 implies that decreasing the dual error by a factor $O(\text{poly}(\kappa_\lambda)c)$ decreases the induced primal error by c. Therefore, applying APCG to the dual and performing the primal and dual mappings yield the theorem. □

2.2 Algorithm

Our Approximate Proximal Point Algorithm (APPA) is given by the following Algorithm 1.

The central goal of this section is to prove the following lemma, which guarantees a geometric convergence rate for the iterates produced in this manner

[1] When the oracle is a randomized algorithm, we require that expected error is the same, *i.e.* that the solution be ϵ-approximate in expectation.

[2] AP-SDCA could likely also serve as a primal oracle with the same guarantees. However, the results in [Shalev-Shwartz and Zhang(2014)] are stated assuming initial primal and dual variables are zero. It is not directly clear how one can provide a generic relative decrease in error from this specific initial primal-dual pair.

Lemma 2.3 (Contraction in APPA)**.** *For any $c' \in (0,1)$, $x \in \mathbb{R}^d$, and possibly randomized primal $(\frac{\lambda+\mu}{c'\mu}, \lambda)$-oracle $\mathcal{P}$ (possibly randomized) we have*

$$\mathbb{E}[F(\mathcal{P}(x))] - F^{opt} \leq \frac{\lambda + c'\mu}{\lambda + \mu}\left(F(x) - F^{opt}\right). \tag{7}$$

This lemma immediately implies the following running-time bounds for APPA.

Theorem 2.4 (Un-regularizing in APPA)**.** *Given a primal $(\frac{2(\mu+\lambda)}{\mu}, \lambda)$-oracle $\mathcal{P}$, Algorithm 1 minimizes the general ERM problem (2) to within accuracy ϵ in time $O(\mathcal{T}_{\mathcal{P}}\lceil\lambda/\mu\rceil \log(\epsilon_0/\epsilon))$.*[3]

Combining Theorem 2.4 and Theorem 2.2 immediately yields our desired running time for solving (1).

Corollary 2.5. *Instantiating Algorithm 1 with the Theorem 2.2 as the primal oracle and taking $\lambda = \mu$ yields the running time of $\widetilde{O}(nd\sqrt{\kappa}\log(\epsilon_0/\epsilon))$ for solving (1).*

2.3 Analysis

This section gives a proof of Lemma 2.3. Throughout, no assumption is made on F aside from μ-strong convexity. Namely, we need not have F be smooth or at all differentiable.

First, we consider the effect of an exact inner minimizer. Namely, we prove the following lemma relating the minimum of the inner problem $f_{s,\lambda}$ to F^{opt}.

Lemma 2.6 (Relationship between minima)**.** *For all $s \in \mathbb{R}^d$ and $\lambda \geq 0$*

$$f_{s,\lambda}^{opt} - F^{opt} \leq \frac{\lambda}{\mu + \lambda}\left(F(s) - F^{opt}\right).$$

Proof. Let $x^{\text{opt}} = \operatorname{argmin}_x F(x)$ and for all $\alpha \in [0,1]$ let $x_\alpha = (1-\alpha)s + \alpha x^{\text{opt}}$. The μ-strong convexity of F implies that, for all $\alpha \in [0,1]$,

$$F(x_\alpha) \leq (1-\alpha)F(s) + \alpha F(x^{\text{opt}}) - \frac{\alpha(1-\alpha)\mu}{2}\|s - x^{\text{opt}}\|_2^2.$$

Consequently, by the definition of $f_{s,\lambda}^{\text{opt}}$,

$$f_{s,\lambda}^{\text{opt}} \leq F(x_\alpha) + \frac{\lambda}{2}\|x_\alpha - s\|_2^2 \leq (1-\alpha)F(s) + \alpha F(x^{\text{opt}}) - \frac{\alpha(1-\alpha)\mu}{2}\|s - x^{\text{opt}}\|_2^2 + \frac{\lambda\alpha^2}{2}\|s - x^{\text{opt}}\|_2^2$$

Choosing $\alpha = \frac{\mu}{\mu+\lambda}$ yields the result. □

This immediately implies contraction for the exact PPA, as it implies that in every iteration of PPA the error in F decreases by a multiplicative $\lambda/(\lambda+\mu)$. Using this we prove Lemma 2.3.

Proof of Lemma 2.3. Let $x' = P(x)$. By definition of primal oracle P we have

$$f_{x,\lambda}(x') - f_{x,\lambda}^{\text{opt}} \leq \frac{c'\mu}{\lambda + \mu}\left(f_{x,\lambda}(x) - f_{x,\lambda}^{\text{opt}}\right).$$

Combining this and Lemma 2.6 we have

$$f_{x,\lambda}(x') - F^{\text{opt}} \leq \frac{c'\mu}{\lambda + \mu}\left(f_{x,\lambda}(x) - f_{x,\lambda}^{\text{opt}}\right) + \frac{\lambda}{\mu + \lambda}\left(F(x) - F^{\text{opt}}\right)$$

Using that clearly for all z we have $F(z) \leq f_{x,\lambda}(z)$ we see that $F(x') \leq f_{x,\lambda}(x')$ and $F^{\text{opt}} \leq f_{x,\lambda}^{\text{opt}}$. Combining with the fact that $f_{x,\lambda}(x) = F(x)$ yields the result. □

[3] When the oracle is a randomized algorithm, the expected accuracy is at most ϵ.

Algorithm 2 Accelerated APPA

```
**Input:** $x^{(0)} \in \mathbb{R}^d$, $\mu > 0$, $\lambda > 2\mu$
**Input:** primal $(4\rho^{3/2}, \lambda)$-oracle $\mathcal{P}$, where $\rho = \frac{\mu+2\lambda}{\mu}$
Define $\zeta = \frac{2}{\mu} + \frac{1}{\lambda}$
$v^{(0)} \leftarrow x^{(0)}$
**for** $t = 0, \ldots, T-1$ **do**
    $y^{(t)} \leftarrow \left(\frac{1}{1+\rho^{-1/2}}\right) x^{(t)} + \left(\frac{\rho^{-1/2}}{1+\rho^{-1/2}}\right) v^{(t)}$
    $x^{(t+1)} \leftarrow \mathcal{P}(y^{(t)})$
    $g^{(t)} \leftarrow \lambda(y^{(t)} - x^{(t+1)})$
    $v^{(t+1)} \leftarrow (1-\rho^{-1/2})v^{(t)} + \rho^{-1/2}\left[y^{(t)} - \zeta g^{(t)}\right]$
**end for**
**OUTPUT:** $x^{(T)}$
```

3 Accelerated APPA

In this section we show how generically accelerate the APPA algorithm of Section 2. Accelerated APPA (Algorithm 2) uses inner minimizers more efficiently, but requires a smaller minimization factor when compared to APPA. The algorithm and its analysis immediately prove Theorem 1.1 and in turn yield another means by which we achieve the accelerated running time guarantees for solving (1).

We first present the algorithm and state its running time guarantees (Section 3.1), then prove the guarantees as part of analysis (Section 3.2).

3.1 Algorithm

Our accelerated APPA algorithm is given by Algorithm 2. In every iteration it still makes a single call to a primal oracle, but rather than requiring a fixed constant minimization the minimization factor depends polynomial on the ratio of λ and μ.

The central goal is to prove the following theorem regarding the running time of APPA.

Theorem 3.1 (Un-regularizing in Accelerated APPA)**.** *Given a primal* $(4(\frac{2\lambda+\mu}{\mu})^{3/2}, \lambda)$*-oracle* $\mathcal{P}$ *for* $\lambda \geq 2\mu$*, Algorithm 2 minimizes the general ERM problem* (2) *to within accuracy* ϵ *in time* $O(\mathcal{T}_{\mathcal{P}}\sqrt{\lceil \lambda/\mu \rceil}\log(\epsilon_0/\epsilon))$*.*

This theorem is essentially a restatement of Theorem 1.1 and by instantiating it with Theorem 2.1 we obtain the following.

Corollary 3.2. *Instantiating Theorem 3.1 with SVRG [Johnson and Zhang(2013)] as the primal oracle and taking* $\lambda = 2\mu + LR^2$ *yields the running time bound* $\widetilde{O}(nd\sqrt{\kappa}\log(\epsilon_0/\epsilon))$ *for the general ERM problem* (2)*.*

3.2 Analysis

Here we establish the convergence rate of Algorithm 2, Accelerated APPA, and prove Theorem 3.1. Note that as in Section 2 the results in this section use nothing about the structure of F other than strong convexity and thus they apply to the general ERM problem (2).

We remark that aspects of the proofs in this section bear resemblance to the analysis in [Shalev-Shwartz and Zhang(2014)], which achieves similar results in a more specialized setting.

Our proof is split into the following parts.

- In Lemma 3.3 we show that applying a primal oracle to the inner minimization problem gives us a quadratic lower bound on $F(x)$.
- In Lemma 3.4 we use this lower bound to construct a series of lower bounds for the main objective function f, and accelerate the APPA algorithm, comprising the bulk of the analysis.
- In Lemma 3.5 we show that the requirements of Lemma 3.4 can be met by using a primal oracle that decreases the error by a constant factor.
- In Lemma 3.6 we analyze the initial error requirements of Lemma 3.4.

The proof of Theorem 3.1 follows immediately from these lemmas.

Lemma 3.3. *For $x_0 \in \mathbb{R}^n$ and $\epsilon > 0$ suppose that x^+ is an ϵ-approximate solution to $f_{x_0,\lambda}$. Then for $\mu' \overset{\text{def}}{=} \mu/2$, $g \overset{\text{def}}{=} \lambda(x_0 - x^+)$, and all $x \in \mathbb{R}^n$ we have*

$$F(x) \geq F(x^+) - \frac{1}{2\mu'}\|g\|^2 + \frac{\mu'}{2}\left\|x - \left(x_0 - \left(\frac{1}{\mu'} + \frac{1}{\lambda}\right)g\right)\right\|_2^2 - \frac{\lambda + 2\mu'}{\mu'}\epsilon.$$

Note that as $\mu' = \mu/2$ we are only losing a factor of 2 in the strong convexity parameter for our lower bound. This allows us to account for errors without sacrificing in our ultimate asymptotic convergence rates.

Proof. Since F is μ-strongly convex clearly $f_{x_0,\lambda}$ is $\mu + \lambda$ strongly convex, by Lemma A.1

$$f_{x_0,\lambda}(x) - f_{x_0,\lambda}(x^{\text{opt}}_{x_0,\lambda}) \geq \frac{\mu+\lambda}{2}\|x - x^{\text{opt}}\|_2^2. \tag{8}$$

By Cauchy-Schwartz and Young's Inequality we know that

$$\frac{\lambda+\mu'}{2}\|x - x^+\|_2^2 \leq \frac{\lambda+\mu'}{2}\left(\|x - x^{\text{opt}}_{x_0,\lambda}\|_2^2 + \|x^{\text{opt}}_{x_0,\lambda} - x^+\|_2^2\right) + \frac{\mu'}{2}\|x - x^{\text{opt}}_{x_0,\lambda}\|_2^2 + \frac{(\lambda+\mu')^2}{2\mu'}\|x^{\text{opt}}_{x_0,\lambda} - x^+\|_2^2,$$

which implies

$$\frac{\mu+\lambda}{2}\|x - x^{\text{opt}}_{x_0,\lambda}\|_2^2 \geq \frac{\lambda+\mu'}{2}\|x - x^+\|_2^2 - \frac{\lambda+\mu'}{\mu'}\cdot\frac{\lambda+\mu}{2}\|x^{\text{opt}}_{x_0,\lambda} - x^+\|_2^2.$$

On the other hand, since $f_{x_0,\lambda}(x^+) \leq f_{x_0,\lambda}(x^{\text{opt}}_{x_0,\lambda}) + \epsilon$ by assumption we have $\frac{\lambda+\mu}{2}\|x^+ - x^{\text{opt}}\|_2^2 \leq \epsilon$ and therefore

$$\begin{aligned}
f_{x_0,\lambda}(x) - f_{x_0,\lambda}(x^+) &\geq f_{x_0,\lambda}(x) - f_{x_0,\lambda}(x^{\text{opt}}_{x_0,\lambda}) - \epsilon \geq \frac{\mu+\lambda}{2}\|x - x^{\text{opt}}_{x_0,\lambda}\|_2^2 - \epsilon \\
&\geq \frac{\lambda+\mu'}{2}\|x - x^+\|_2^2 - \frac{\lambda+\mu'}{\mu'}\cdot\frac{\lambda+\mu}{2}\|x^{\text{opt}}_{x_0,\lambda} - x^+\|_2^2 - \epsilon \\
&\geq \frac{\lambda+\mu'}{2}\|x - x^+\|_2^2 - \frac{\lambda+2\mu'}{\mu'}\epsilon.
\end{aligned}$$

Now since

$$\|x-x^+\|_2^2=\|x-x_0+\frac{1}{\lambda}g\|_2^2=\|x-x_0\|^2+\frac{2}{\lambda}\left\langle g,x-x_0\right\rangle+\frac{1}{\lambda^2}\|g\|_2^2,$$

and using the fact that $f_{x_0,\lambda}(x)=F(x)+\frac{\lambda}{2}\|x-x_0\|_2^2$, we have

$$F(x)\geq F(x^+)+\left[\frac{1}{\lambda}+\frac{\mu'}{2\lambda^2}\right]\|g\|_2^2+\left(1+\frac{\mu'}{\lambda}\right)\left\langle g,x-x_0\right\rangle+\frac{\mu'}{2}\|x-x_0\|^2-\frac{\lambda+2\mu'}{\mu'}\epsilon.$$

The right hand side of the above equation is a quadratic function. Looking at its gradient with respect to x we see that it obtains its minimum when $x=x_0-(\frac{1}{\mu'}+\frac{1}{\lambda})g$ and has a minimum value of $F(x^+)-\frac{1}{2\mu'}\|g\|_2^2-\frac{\lambda+2\mu'}{\mu'}\epsilon$. □

Lemma 3.4. *Suppose that in each iteration t we have $\psi_t\overset{\text{def}}{=}\psi_t^{opt}+\frac{\mu'}{2}\|x-v^{(t)}\|_2^2$ such that $F(x)\geq\psi_t(x)$ for all x. Let $\rho\overset{\text{def}}{=}\frac{\mu'+\lambda}{\mu'}$ for $\lambda\geq 3\mu'$, and let*

- $y^{(t)}\overset{\text{def}}{=}\left(\frac{1}{1+\rho^{-1/2}}\right)x^{(t)}+\left(\frac{\rho^{-1/2}}{1+\rho^{-1/2}}\right)v^{(t)}$,
- $\mathbb{E}[f_{y^{(t)},\lambda}(x^{(t+1)})]-f_{y^{(t)},\lambda}^{opt}\leq\frac{\rho^{-3/2}}{4}(F(x^{(t)})-\psi_t^{opt})$,
- $g^{(t)}\overset{\text{def}}{=}\lambda(y^{(t)}-x^{(t+1)})$,
- $v^{(t+1)}\overset{\text{def}}{=}(1-\rho^{-1/2})v^{(t)}+\rho^{-1/2}\left[y^{(t)}-\left(\frac{1}{\mu'}+\frac{1}{\lambda}\right)g^{(t)}\right]$.

We have

$$\mathbb{E}[F(x^{(t)})-\psi_t^{opt}]\leq\left(1-\frac{\rho^{-1/2}}{2}\right)^t(F(x_0)-\psi_0^{opt}).$$

Proof. Regardless of how $y^{(t)}$ is chosen we know by Lemma 3.3 that for $\gamma=1+\frac{\mu'}{\lambda}$ and all $x\in\mathbb{R}^n$

$$F(x)\geq F(x^{(t+1)})-\frac{1}{2\mu'}\|g^{(t)}\|_2^2+\frac{\mu'}{2}\left\|x-\left(y^{(t)}-\frac{\gamma}{\mu'}g^{(t)}\right)\right\|_2^2-\frac{\lambda+2\mu'}{\mu'}\left(f_{y^{(t)},\lambda}(x^{(t+1)})-f_{y^{(t)},\lambda}^{\text{opt}}\right).\quad(9)$$

Thus, for $\beta=1-\rho^{-1/2}$ we can let

$$\begin{aligned}\psi_{t+1}(x)&\overset{\text{def}}{=}\beta\psi_t(x)+(1-\beta)\left[F(x^{(t+1)})-\frac{1}{2\mu'}\|g^{(t)}\|_2^2+\frac{\mu'}{2}\|x-\left(y^{(t)}-\frac{\gamma}{\mu'}g^{(t)}\right)\|_2^2\right.\\&\quad\left.-\frac{\lambda+2\mu'}{\mu'}(f_{y^{(t)},\lambda}(x^{(t+1)})-f_{y^{(t)},\lambda}^{\text{opt}})\right]\\&=\beta\left[\psi_t^{\text{opt}}+\frac{\mu'}{2}\|x-v^{(t)}\|_2^2\right]+(1-\beta)\left[F(x^{(t+1)})-\frac{1}{2\mu'}\|g^{(t)}\|_2^2+\frac{\mu'}{2}\|x-\left(y^{(t)}-\frac{\gamma}{\mu'}g^{(t)}\right)\|_2^2\right.\\&\quad\left.-\frac{\lambda+2\mu'}{\mu'}(f_{y^{(t)},\lambda}(x^{(t+1)})-f_{y^{(t)},\lambda}^{\text{opt}})\right]\\&=\psi_{t+1}^{\text{opt}}+\frac{\mu'}{2}\|x-v^{(t+1)}\|_2^2.\end{aligned}$$

where in the last line we used Lemma A.3. Again, by Lemma A.3 we know that

$$\begin{aligned}\psi_{t+1}^{\text{opt}} &= \beta\psi_t + (1-\beta)\left(F(x^{(t+1)}) - \frac{1}{2\mu'}\|g^{(t)}\|_2^2 - \frac{\lambda+2\mu'}{\mu'}(f_{y^{(t)},\lambda}(x^{(t+1)}) - f_{y^{(t)},\lambda}^{\text{opt}})\right)\\ &\quad + \beta(1-\beta)\frac{\mu'}{2}\|v^{(t)} - \left(y^{(t)} - \frac{\gamma}{\mu'}g^{(t)}\right)\|_2^2\\ &\geq \beta\psi_t + (1-\beta)F(x^{(t+1)}) - \frac{(1-\beta)^2}{2\mu'}\|g^{(t)}\|_2^2 + \beta(1-\beta)\gamma\left\langle g^{(t)}, v^{(t)} - y^{(t)}\right\rangle\\ &\quad - \frac{(1-\beta)(\lambda+2\mu')}{\mu'}(f_{y^{(t)},\lambda}(x^{(t+1)}) - f_{y^{(t)},\lambda}^{\text{opt}}).\end{aligned}$$

In the second step we used the following fact:

$$-\frac{1-\beta}{2\mu'} + \beta(1-\beta)\frac{\mu'}{2}\cdot\frac{\gamma^2}{\mu'} = \frac{1-\beta}{2\mu'}(-1+\beta\gamma^2) \geq -\frac{(1-\beta)^2}{2\mu'}.$$

Furthermore, expanding the term $\frac{\mu}{2}\|(x-y^{(t)}) + \frac{\gamma}{\mu}g^{(t)}\|_2^2$ and instantiating x with $x^{(t)}$ in (9) yields

$$F(x^{(t+1)}) \leq F(x^{(t)}) - \frac{1}{\lambda}\|g^{(t)}\|_2^2 + \gamma\left\langle g^{(t)}, y^{(t)} - x^{(t)}\right\rangle + \frac{\lambda+2\mu'}{\mu'}(f_{y^{(t)},\lambda}(x^{(t+1)}) - f_{y^{(t)},\lambda}^{\text{opt}}).$$

Consequently we know

$$\begin{aligned}F(x^{(t+1)}) - \psi_{t+1}^{\text{opt}} &\leq \beta[f(x^{(t)}) - \psi_t^{\text{opt}}] + \left[\frac{(1-\beta)^2}{2\mu'} - \frac{\beta}{\lambda}\right]\|g^{(t)}\|_2^2 + \gamma\beta\left\langle g^{(t)}, y^{(t)} - x^{(t)} - (1-\beta)(v^{(t)} - y^{(t)})\right\rangle\\ &\quad + \frac{(\lambda+2\mu')}{\mu'}(f_{y^{(t)},\lambda}(x^{(t+1)}) - f_{y^{(t)},\lambda}^{\text{opt}})\end{aligned}$$

Note that we have chosen $y^{(t)}$ so that the inner product term equals 0, and we choose $\beta = 1-\rho^{-1/2} \geq \frac{1}{2}$ which ensures

$$\frac{(1-\beta)^2}{2\mu'} - \frac{\beta}{\lambda} \leq \frac{1}{2(\mu'+\lambda)} - \frac{1}{2\lambda} \leq 0.$$

Also, by assumption we know $\mathbb{E}[f_{y^{(t)},\lambda}(x^{(t+1)}) - f_{y^{(t)},\lambda}^{\text{opt}}] \leq \frac{\rho^{-3/2}}{4}(f(x^{(t)}) - \psi_t^{\text{opt}})$, which implies

$$\mathbb{E}[F(x^{(t+1)}) - \psi_{t+1}^{\text{opt}}] \leq \left(\beta + \frac{(\lambda+2\mu')}{\mu'}\cdot\frac{\rho^{-3/2}}{4}\right)(F(x^{(t)}) - \psi_t^{\text{opt}}) \leq (1-\rho^{-1/2}/2)(F(x^{(t)}) - \psi_t^{\text{opt}}).$$

In the final step we are using the fact that $\frac{\lambda+2\mu'}{\mu'} \leq 2\rho$ and $\rho \geq 1$. □

Lemma 3.5. *Under the setting of Lemma 3.4, we have $f_{y^{(t)},\lambda}(x^{(t)}) - f_{y^{(t)},\lambda}^{opt} \leq F(x^{(t)}) - \psi_t^{opt}$. In particular, in order to achieve $\mathbb{E}[f_{y^{(t)},\lambda}(x^{(t+1)})] \leq \frac{\rho^{-3/2}}{8}(F(x^{(t)}) - \psi_t^{opt})$ we only need an oracle that shrinks the function error by a factor of $\frac{\rho^{-3/2}}{8}$ (in expectation).*

Proof. We know

$$f_{y^{(t)},\lambda}(x^{(t)}) - f(x^{(t)}) = \frac{\lambda}{2}\|x^{(t)} - y^{(t)}\|_2^2 = \frac{\lambda}{2}\cdot\frac{\rho^{-1}}{(1+\rho^{-1/2})^2}\|x^{(t)} - v^{(t)}\|_2^2.$$

We will try to show the lower bound $f^{\text{opt}}_{y^{(t)},\lambda}$ is larger than ψ^{opt}_t by the same amount. This is because for all x we have

$$f_{y^{(t)},\lambda}(x) = F(x) + \frac{\lambda}{2}\|x - y^{(t)}\|_2^2 \geq \psi^{\text{opt}}_t + \frac{\mu'}{2}\|x - v^{(t)}\|_2^2 + \frac{\lambda}{2}\|x - y^{(t)}\|_2^2.$$

The right hand side is a quadratic function, whose optimal point is at $x = \frac{\mu' v^{(t)} + \lambda y^{(t)}}{\mu' + \lambda}$ and whose optimal value is equal to

$$\psi^{\text{opt}}_t + \frac{\lambda}{2}\left(\frac{\mu'}{\mu' + \lambda}\right)^2 \|v^{(t)} - y^{(t)}\|_2^2 + \frac{\mu'}{2}\left(\frac{\lambda}{\mu + \lambda}\right)^2 \|v^{(t)} - y^{(t)}\|_2^2 = \psi^{\text{opt}}_t + \frac{\mu'\lambda}{2(\mu' + \lambda)} \cdot \frac{1}{(1 + \rho^{-1/2})^2}\|x^{(t)} - v^{(t)}\|_2^2.$$

By definition of ρ^{-1}, we know $\frac{\mu'\lambda}{2(\mu'+\lambda)} \cdot \frac{1}{(1+\rho^{-1/2})^2}\|x^{(t)} - v^{(t)}\|_2^2$ is exactly equal to $\frac{\lambda}{2} \cdot \frac{\rho^{-1}}{(1+\rho^{-1/2})^2}\|x^{(t)} - v^{(t)}\|_2^2$, therefore $f_{y^{(t)},\lambda}(x^{(t)}) - f^{\text{opt}}_{y^{(t)},\lambda} \leq F(x^{(t)}) - \psi^{\text{opt}}_t$. □

Remark In the next lemma we show that moving to the regularized problem has the same effect on the primal function value and the lower bound. This is a result of the choice of β in the proof of Lemma 3.4. However, this does not mean that the choice of β is very fragile. We can choose any β' that is between the current β and 1; the effect on this lemma will be that the increase in primal function becomes smaller than the increase in the lower bound (so the lemma continues to hold).

Lemma 3.6. *Let $\psi^{opt}_0 = F(x^{(0)}) - \frac{\lambda + 2\mu'}{\mu'}(F(x^{(0)}) - f^{opt})$, and $v^{(0)} = x^{(0)}$, then $\psi_0 \overset{\text{def}}{=} \psi^{opt}_0 + \frac{\mu'}{2}\|x - v_0\|^2$ is a valid lower bound for F. In particular when $\lambda = LR^2$ then $F(x^{(0)}) - \psi^{opt}_0 \leq 2\kappa(F(x^{(0)}) - f^{opt})$.*

Proof. This lemma is a direct corollary of Lemma 3.3 with $x^+ = x^{(0)}$. □

4 Dual APPA

In this section we develop Dual APPA (Algorithm 3), a natural approximate proximal point algorithm that operates entirely in the regularized ERM dual. Our focus here is on theoretical properties of Dual APPA .

We first present an abstraction for dual-based inner minimizers (Section 4.1), then present the algorithm (Section 4.2), and finally step through its runtime analysis (Section 4.3).

4.1 Approximate dual oracles

Our primary goal in this section is to quantify how much objective function progress an algorithm needs to make in the dual problem, $g_{s,\lambda}$ (See Section 1.1) in order to ensure primal progress at a rate similar to that in APPA (Algorithm 1).

Here, similar to Section 2.1, we formally define our requirements for an approximate dual-based inner dual minimize. In particular, we use the following notion of dual oracle.

$:=$ An algorithm $\mathcal{D}$ is a *dual* (c, λ)*-oracle* if, given $s \in \mathbb{R}^d$ and $y \in \mathbb{R}^n$, it outputs $\mathcal{D}(s, y)$ that is a $([g_{s,\lambda}(y) - g^{\text{opt}}_{s,\lambda}]/c)$-approximate minimizer of $g_{s,\lambda}$ in time $\mathcal{T}_{\mathcal{D}}$.[4]

Dual based algorithms for regularized ERM and variants of coordinate descent typically can be used as such a dual oracle. In particular we note that APCG is such a dual oracle.

[4] As in the primal oracle definition, when the oracle is a randomized algorithm, we require that its output be an expected ϵ-approximate solution.

Algorithm 3 Dual APPA

```
Input: $x^{(0)} \in \mathbb{R}^d$, $\lambda > 0$
Input: dual $(\sigma, \lambda)$-oracle $\mathcal{D}$                    (see Theorem 4.2 for $\sigma$)
$y^{(0)} \leftarrow \widehat{y}(x^{(0)})$
for $t = 1, \ldots, T$ do
  $y^{(t)} \leftarrow \mathcal{D}(x^{(t-1)}, y^{(t-1)})$
  $x^{(t)} \leftarrow \widehat{x}_{x^{(t-1)},\lambda}(y^{(t)})$
end for
OUTPUT: $x^{(T)}$
```

Theorem 4.1 (APCG as a dual oracle)**.** *APCG [Lin et al.(2014)] is a dual (c, λ)-oracle with runtime complexity $\mathcal{T}_{\mathcal{D}} = \widetilde{O}(nd\sqrt{\kappa_\lambda} \log c)$.*[5]

4.2 Algorithm

Our dual APPA is given by the following Algorithm 3.

Dual APPA (Algorithm 3) repeatedly queries a dual oracle while producing primal iterates via the dual-to-primal mapping (5) along the way. We show that it obtains the following running time bound:

Theorem 4.2 (Un-regularizing in Dual APPA)**.** *Given a dual (σ, λ)-oracle $\mathcal{D}$, where*

$$\sigma \geq 80 n^2 \kappa_\lambda^2 \max\{\kappa, \kappa_\lambda\} \lceil \lambda/\mu \rceil$$

Algorithm 3 minimizes the ERM problem (1) *to within accuracy ϵ in time $\widetilde{O}(\mathcal{T}_{\mathcal{D}} \lceil \lambda/\mu \rceil \log(\epsilon_0/\epsilon))$.*[6]

Combining Theorem 4.2 and Theorem 4.1 immediately yields another way to achieve our desired running time for solving (1).

Corollary 4.3. *Instantiating Theorem 4.2 with Theorem 4.1 as the dual oracle and taking $\lambda = \mu$ yields the running time bound $\widetilde{O}(nd\sqrt{\kappa} \log(\epsilon_0/\epsilon))$.*

While both this result and the results in Section 2 show that APCG can be used to achieve our fastest running times for solving (1), note that the algorithms they suggest are in fact different. In every invocation of APCG in Algorithm 1, we need to explicitly compute both the primal-to-dual and dual-to-primal mappings (in $O(nd)$ time). However, here we only need to compute the primal-to-dual mapping once upfront, in order to initialize the algorithm. Every subsequent invocation of APCG then only requires a single dual-to-primal mapping computation, which can often be streamlined. From a practical viewpoint, this can be seen as a natural "warm start" scheme for the dual-based inner minimizer.

4.3 Analysis

Here we proves Theorem 4.2. We begin by bounding the error of the dual regularized ERM problem when the center of regularization changes. This characterizes the initial error at the beginning of each Dual APPA iteration.

[5] As in Theorem 2.2, AP-SDCA could likely also serve as a dual oracle with the same guarantees, provided it is modified to allow for the more general primal-dual initialization.

[6] As in Theorem 2.4, when the oracle is a randomized algorithm, the expected accuracy is at most ϵ.

Lemma 4.4 (Dual error after re-centering.)**.** *For all $y \in \mathbb{R}^n$, $x \in \mathbb{R}^d$, and $x' = \widehat{x}_x(y)$ we have*

$$g_{x',\lambda}(y) - g^{opt}_{x',\lambda} \leq 2(g_{x,\lambda}(y) - g^{opt}_{x,\lambda}) + 4n\kappa \left[F(x') - F^{opt} + F(x) - F^{opt}\right]$$

In other words, the dual error $g_{s,\lambda}(y) - g^{\mathrm{opt}}_{s,\lambda}$ is bounded across a re-centering step by multiples of previous sub-optimality measurements (namely, dual error and gradient norm).

Proof. By the definition of $g_{x,\lambda}$ and x' we have, for all z,

$$g_{x',\lambda}(z) = G(z) + \frac{1}{2\lambda}\|A^{\mathsf{T}}z\|^2 - x'^{\mathsf{T}}A^{\mathsf{T}}z = g_{x,\lambda}(z) - (x'-x)^{\top}A^{\mathsf{T}}z = g_{x,\lambda}(z) + \frac{1}{\lambda}y^{\mathsf{T}}AA^{\mathsf{T}}z \ .$$

Furthermore, since g is $\frac{1}{L}$-strongly convex we can invoke Lemma A.2 obtaining

$$g_{x',\lambda}(y) - g^{\mathrm{opt}}_{x',\lambda} \leq 2\left[g_{x,\lambda}(y) - g^{\mathrm{opt}}_{x',\lambda}\right] + L\left\|\frac{1}{\lambda}AA^{\mathsf{T}}y\right\|_2^2 .$$

Since each row of A has ℓ_2 norm at most R we know that $\|Az\|_2^2 \leq nR^2\|z\|_2^2$ and we know that by definition $A^{\mathsf{T}}y = \lambda(x - x')$. Combining these yields

$$g_{x',\lambda}(y) - g^{\mathrm{opt}}_{x',\lambda} \leq 2\left[g_{x,\lambda}(y) - g^{\mathrm{opt}}_{x',\lambda}\right] + nLR^2\|x - x'\|_2^2.$$

Finally, since F is μ-strongly convex, by Lemma A.1, we have

$$\frac{1}{2}\|x - x'\|_2^2 \leq \|x' - x^{\mathrm{opt}}\|_2^2 + \|x - x^{\mathrm{opt}}\|_2^2 \leq \frac{2}{\mu}\left[F(x') - F^{\mathrm{opt}} + F(x) - F^{\mathrm{opt}}\right] \ .$$

Combining and recalling the definition of κ yields the result. □

The following lemma establishes the rate of convergence of the primal iterates $\{x^{(t)}\}$ produced over the course of Dual APPA, and in turn implies Theorem 4.2.

Lemma 4.5 (Convergence rate of Dual APPA)**.** *Let $c' \in (0,1)$ be arbitrary and suppose that $\sigma \geq (40/c')n^2\kappa_\lambda^2 \max\{\kappa, \kappa_\lambda\}\lceil \lambda/\mu \rceil$ in Dual APPA (Algorithm 3). Then in every iteration $t \geq 1$ of Dual APPA (Algorithm 3) the following invariants hold:*

$$F(x^{(t-1)}) - F^{opt} \leq \left(\frac{\lambda + c'\mu}{\lambda + \mu}\right)^{t-1}\left(F(x^{(0)}) - F^{opt}\right), \quad \textit{and} \tag{10}$$

$$g_{x^{(t-1)},\lambda}(y^{(t)}) - g^{opt}_{x^{(t-1)},\lambda} \leq \left(\frac{\lambda + c'\mu}{\lambda + \mu}\right)^{t-1}\left(F(x^{(0)}) - F^{opt}\right). \tag{11}$$

Proof. For notational convenience we let $r \overset{\mathrm{def}}{=} (\frac{\lambda + c'\mu}{\lambda+\mu})$, $g_t \overset{\mathrm{def}}{=} g_{x^{(t)},\lambda}$, $f_t \overset{\mathrm{def}}{=} f_{x^{(t)},\lambda}$, and $\epsilon_t \overset{\mathrm{def}}{=} F(x^{(t)}) - F^{\mathrm{opt}}$ for all $t \geq 0$. Thus, we wish to show that $\epsilon_{t-1} \leq r^{t-1}\epsilon_0$ (equivalent to (11)) and we wish to show that $g_{t-1}(y^{(t)}) - g^{\mathrm{opt}}_{t-1} \leq r^{t-1}\epsilon_0$ (equivalent to (10)) for all $t \geq 1$.

By definition of a dual oracle we have, for all $t \geq 1$,

$$g_{t-1}(y^{(t)}) - g^{\mathrm{opt}}_{t-1} \leq \frac{1}{\sigma}\left[g_{t-1}(y_{t-1}) - g^{\mathrm{opt}}_{t-1}\right], \tag{12}$$

by Lemma B.1 we have, for all $t \geq 1$,

$$f_{t-1}(x^{(t)}) - f_{t-1}^{\text{opt}} \leq 2n^2\kappa_\lambda^2 \left[g_{t-1}(y^{(t)}) - g_{t-1}^{\text{opt}}\right], \tag{13}$$

by Lemma 4.4 we know

$$g_t(y^{(t)}) - g_t^{\text{opt}} \leq 2\left[g_{t-1}(y^{(t)}) - g_t^{\text{opt}}\right] + 4n\kappa(\epsilon_t + \epsilon_{t-1}), \tag{14}$$

and by Lemma 2.6 we know that for all $t \geq 1$

$$f_{t-1}^{\text{opt}} - F^{\text{opt}} \leq \frac{\lambda}{\mu + \lambda}\epsilon_{t-1} \tag{15}$$

Furthermore, by Corollary B.3, the definition of $y^{(0)}$, and the facts that $f_0(x^{(0)}) = F(x^{(0)})$ and $f_t(z) \geq F(z)$ we have

$$g_0(y^{(0)}) - g_0^{\text{opt}} \leq 2\kappa_\lambda \left(f_0(x^{(0)}) - f_0^{\text{opt}}\right) \leq 2\kappa_\lambda \left(F(x^{(0)}) - F^{\text{opt}}\right) = 2\kappa_\lambda\epsilon_0 \tag{16}$$

We show that combining these and applying strong induction on t yields the desired result.

We begin with our base cases. When $t = 1$ the invariant (11) holds immediately by definition. Furthermore, when $t = 1$ we see that the invariant (10) holds, since $\sigma \geq 2\kappa_\lambda$ and

$$g_0(y^{(1)}) - g_0^{\text{opt}} \leq \frac{1}{\sigma}(g_0(y^{(0)}) - g_0^{\text{opt}}) \leq \frac{2\kappa_\lambda}{\sigma}\left(f_0(x^{(0)}) - f_0^{\text{opt}}\right) \leq \frac{2\kappa_\lambda}{\sigma}\epsilon_0, \tag{17}$$

were we used (12) and (16) respectively. Finally we show that invariant (11) holds for $t = 2$:

$$\begin{aligned}
F(x^{(1)}) - F^{\text{opt}} &\leq f_0(x^{(1)}) - f_0^{\text{opt}} + f_0^{\text{opt}} - F^{\text{opt}} && \text{(Since } F(z) \leq f_t(z) \text{ for all } t, z\text{)} \\
&\leq 2n^2\kappa_\lambda^2(g_0(y^{(1)}) - g_0^{\text{opt}}) + \frac{\lambda}{\mu + \lambda}\epsilon_0 && \text{(Equations (13) and (15))} \\
&\leq \left(\frac{4n^2\kappa_\lambda^3}{\sigma} + \frac{\lambda}{\mu + \lambda}\right)\epsilon_0 && \text{(Equation (17))} \\
&\leq r\epsilon_0 && \text{(Since } \sigma \geq 4n\kappa_\lambda^3/(c'\lambda/(\mu + \lambda))\text{)}
\end{aligned}$$

Now consider $t \geq 3$ for the second invariant (11). We show this holds assuming the invariants hold for all smaller t.

$$\begin{aligned}
F(x^{(t-1)}) - F^{\text{opt}} &\leq f_{t-2}(x^{(t-1)}) - f_{t-2}^{\text{opt}} + f_{t-2}^{\text{opt}} - F^{\text{opt}} && \text{(Since } F(z) \leq f_t(z) \text{ for all } t, z\text{)} \\
&\leq 2n^2\kappa_\lambda^2(g_{t-2}(y_{t-1}) - g_{t-2}^{\text{opt}}) + \frac{\lambda}{\mu + \lambda}\epsilon_{t-2} && \text{(Equations (13) and (15))} \\
&\leq \frac{2n^2\kappa_\lambda^2}{\sigma}\left(g_{t-2}(y_{t-2}) - g_{t-2}^{\text{opt}}\right) + \frac{\lambda}{\mu + \lambda}\epsilon_{t-2} && \text{(Equation (12))}
\end{aligned}$$

Furthermore,

$$\begin{aligned}
g_{t-2}(y_{t-2}) - g_{t-2}^{\text{opt}} &\leq 2(g_{t-3}(y_{t-2}) - g_{t-3}^{\text{opt}}) + 4n\kappa\left[\epsilon_{t-2} + \epsilon_{t-3}\right] && \text{(Equation (17))} \\
&\leq \left(2r^{t-2} + 4n\kappa(r^{t-1} + r^{t-2})\right)\epsilon_0 && \text{(Inductive hypothesis)} \\
&\leq 10n\kappa r^{t-1}\epsilon_0 && (r \leq 1 \text{ and } \kappa \geq 1)
\end{aligned}$$

Since $\sigma \geq 20n^2\kappa_\lambda^2\kappa/(c'\lambda/(\mu+\lambda))$ combining yields that

$$\frac{2n^2\kappa_\lambda^2}{\sigma}\left(g_{t-2}(y_{t-2}) - g_{t-2}^{\text{opt}}\right) \leq \frac{c'\mu}{\mu+\lambda} r^{t-1}\epsilon_0$$

and the result follows by the inductive hypothesis on ϵ_{t-2}.

Finally we show that invariant (10) holds for any $t \geq 2$ given that it holds for all smaller t and invariant (11) holds for that t and all smaller t.

$$\begin{aligned}
g_{t-1}(y^{(t)}) - g_{t-1}^{\text{opt}} &\leq \frac{1}{\sigma}(g_{t-1}(y_{t-1}) - g_{t-1}^{\text{opt}}) && \text{(Definition dual oracle.)}\\
&\leq \frac{1}{\sigma}\left[2(g_{x^{(t-2)}}(y_{t-1}) - g_{x^{(t-2)}}^{\text{opt}}) + 4n\kappa\left[\epsilon_{t-1} + \epsilon_{t-2}\right]\right] && \text{(Equation (14))}\\
&\leq \frac{1}{\sigma}\left[2r^{t-1} + 4n\kappa\left[r^t + r^{t-1}\right]\right]\epsilon_0 && \text{(Inductive hypothesis)}\\
&\leq r^{t-1}\epsilon_0 && (\sigma \geq 8n\kappa)
\end{aligned}$$

The result then follows by induction. □

5 Conclusion

We have developed a family of accelerated stochastic algorithms that minimize sums of convex functions, with a specific focus on empirical risk minimization (ERM) and linear least-squares regression. Our algorithms outperform existing methods in terms of running time across a broad spectrum of problem settings.

To achieve this, we introduced a framework rooted in the classical *proximal point algorithm*, which systematically reduces the minimization of strongly convex functions to the approximate minimization of their regularized counterparts. Through this approach, we accelerate the fastest stochastic algorithms in a black-box manner, ensuring that the benefits of large regularization terms are exploited without introducing bias to the original problem.

Empirically, our algorithms demonstrate stability and robustness, offering practical benefits in large-scale optimization tasks. Both theoretically and in practice, the presented methods effectively balance the trade-offs between acceleration, stability, and computational efficiency, providing a powerful tool for addressing high-dimensional convex optimization challenges.

References

[Bottou and Bousquet(2008)] L. Bottou and O. Bousquet. The tradeoffs of large scale learning. In *Advances in Neural Information Processing Systems (NIPS)*, 2008.

[Boyd et al.(2011)] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2011.

[Cohen et al.(2015)] M. B. Cohen, Y. T. Lee, C. Musco, C. Musco, R. Peng, and A. Sidford. Uniform sampling for matrix approximation. In *Innovations in Theoretical Computer Science (ITCS)*, 2015.

[Defazio et al.(2014)] 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 (NIPS)*, 2014.

[Guler(1992)] O. Guler. New proximal point algorithms for convex minimization. *SIAM Journal on Optimization*, 2(4):649–664, 1992.

[Johnson and Zhang(2013)] R. Johnson and T. Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems (NIPS)*, 2013.

[Lee and Sidford(2013)] Y. T. Lee and A. Sidford. Efficient accelerated coordinate descent methods and faster algorithms for solving linear systems. In *Foundations of Computer Science (FOCS)*, 2013.

[Li et al.(2013)] M. Li, G. L. Miller, and R. Peng. Iterative row sampling. In *Foundations of Computer Science (FOCS)*, 2013.

[Lin et al.(2015)] H. Lin, J. Mairal, and Z. Harchaoui. A universal catalyst for first-order optimization. *arXiv*, 2015.

[Lin et al.(2014)] Q. Lin, Z. Lu, and L. Xiao. An accelerated proximal coordinate gradient method. In *Advances in Neural Information Processing Systems (NIPS)*, 2014.

[Needell et al.(2014)] D. Needell, N. Srebro, and R. Ward. Stochastic gradient descent, weighted sampling, and the randomized kaczmarz algorithm. In *Advances in Neural Information Processing Systems (NIPS)*, 2014.

[Nelson and Nguyen(2013)] J. Nelson and H. L. Nguyen. OSNAP: Faster numerical linear algebra algorithms via sparser subspace embeddings. In *Foundations of Computer Science (FOCS)*, 2013.

[Nesterov(1983)] Y. Nesterov. A method of solving a convex programming problem with convergence rate $O(1/k^2)$. *Soviet Mathematics Doklady*, 27(2):372–376, 1983.

[Nesterov(2004)] Y. Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*. Springer, 2004.

[Parikh and Boyd(2014)] N. Parikh and S. Boyd. Proximal algorithms. *Foundations and Trends in Optimization*, 1(3):123–231, 2014.

[Rahimi and Recht(2007)] A. Rahimi and B. Recht. Random features for large-scale kernel machines. In *Advances in Neural Information Processing Systems (NIPS)*, 2007.

[Rockafellar(1976)] R. T. Rockafellar. Monotone operators and the proximal point algorithm. *SIAM Journal on Control and Optimization*, 14(5):877–898, 1976.

[Roux et al.(2012)] N. L. Roux, M. Schmidt, and F. Bach. A stochastic gradient method with an exponential convergence rate for finite training sets. In *Advances in Neural Information Processing Systems (NIPS)*, 2012.

[Shalev-Shwartz and Zhang(2013)] S. Shalev-Shwartz and T. Zhang. Stochastic dual coordinate ascent methods for regularized loss minimization. *Journal of Machine Learning Research (JMLR)*, 14:567–599, 2013.

[Shalev-Shwartz and Zhang(2014)] S. Shalev-Shwartz and T. Zhang. Accelerated proximal stochastic dual coordinate ascent for regularized loss minimization. *Mathematical Programming*, pages 1–41, 2014.

[Strohmer and Vershynin(2009)] T. Strohmer and R. Vershynin. A randomized kaczmarz algorithm with exponential convergence. *Journal of Fourier Analysis and Applications*, 15:262–278, 2009.

[Williams(2012)] V. V. Williams. Multiplying matrices faster than Coppersmith-Winograd. In *Symposium on Theory of Computing (STOC)*, 2012.

[Xiao and Zhang(2014)] L. Xiao and T. Zhang. A proximal stochastic gradient method with progressive variance reduction. *SIAM Journal on Optimization*, 24(4):2057–2075, 2014.

A Technical lemmas

In this section we provide several stand-alone technical lemmas we use throughout the paper. First we provide Lemma A.1 some common inequalities regarding smooth or strongly convex functions, then Lemma A.2 which shows the effect of adding a linear term to a convex function, and then Lemma A.3 a small technical lemma regarding convex combinations of quadratic functions.

Lemma A.1 (Standard bounds for smooth, strongly convex functions)**.** *Let $f : \mathbb{R}^k \to \mathbb{R}$ be differentiable function that obtains its minimal value at x^{opt}.*

If f is L-smooth then for all $x \in \mathbb{R}^k$

$$\frac{1}{2L}\|\nabla f(x)\|_2^2 \leq f(x) - f(x^{opt}) \leq \frac{L}{2}\|x - x^{opt}\|_2^2 \enspace .$$

If f is μ-strongly convex the for all $x \in \mathbb{R}^k$

$$\frac{\mu}{2}\|x - x^{opt}\|_2^2 \leq f(x) - f(x^{opt}) \leq \frac{1}{2\mu}\|\nabla f(x)\|_2^2 \enspace .$$

Proof. Apply the definition of smoothness and strong convexity at the points x and x^{opt} and minimize the resulting quadratic form. □

Lemma A.2. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a μ-strongly convex function and for all $a, x \in \mathbb{R}^n$ let $f_a(x) = f(x) + a^\top x$. Then*

$$f_a(x) - f_a^{opt} \leq 2(f(x) - f^{opt}) + \frac{1}{\mu}\|a\|_2^2$$

Proof. [7] Let $x^{\text{opt}} = \text{argmin}_x f(x)$. Since f is μ-strongly convex by Lemma A.1 we have $f(x) \geq f(x^{\text{opt}}) + \frac{\mu}{2}\|x - x^{\text{opt}}\|_2^2$ for all x. Consequently, for all x

$$f_a^{\text{opt}} \geq f(x) + a^\top x \geq f(x^{\text{opt}}) + \frac{\mu}{2}\|x - x^{\text{opt}}\|_2^2 + a^\top x \geq f_a(x^{\text{opt}}) + a^\top(x - x^{\text{opt}}) + \frac{\mu}{2}\|x - x^{\text{opt}}\|_2^2$$

Minimizing with respect to x yields that $f_a^{\text{opt}} \geq f_a(x^{\text{opt}}) - \frac{1}{2\mu}\|a\|_2^2$. Consequently, by Cauchy Schwarz, and Young's Inequality we have

$$\begin{aligned} f_a(x) - f_a^{\text{opt}} &\leq f(x) - f^{\text{opt}} + a^\top(x - x^{\text{opt}}) + \frac{1}{2\mu}\|a\|_2^2 && (18) \\ &\leq f(x) - f^{\text{opt}} + \frac{1}{2\mu}\|a\|_2^2 + \frac{\mu}{2}\|x - x^{\text{opt}}\|_2^2 + \frac{1}{2\mu}\|a\|_2^2 && (19) \end{aligned}$$

Applying A.1 again yields the result. □

Lemma A.3. *Suppose that for all x we have*

$$f_1(x) \stackrel{\text{def}}{=} \psi_1 + \frac{\mu}{2}\|x - v_1\|_2^2 \text{ and } f_2(x) = \psi_2 + \frac{\mu}{2}\|x - v_2\|_2^2$$

then

$$\alpha f_1(x) + (1 - \alpha) f_2(x) = \psi_\alpha + \frac{\mu}{2}\|x - v_\alpha\|_2^2$$

where

$$v_\alpha = \alpha v_1 + (1 - \alpha) v_2 \quad \text{and} \quad \psi_\alpha = \alpha\psi_1 + (1 - \alpha)\psi_2 + \frac{\mu}{2}\alpha(1 - \alpha)\|v_1 - v_2\|_2^2$$

[7] Note we could have also proved this by appealing to the gradient of f and Lemma A.1, however the proof here holds even if f is not differentiable.

Proof. Setting the gradient of $\alpha f_1(x) + (1-\alpha) f_2(x)$ to 0 we know that v_α must satisfy

$$\alpha\mu\,(v_\alpha - v_1) + (1-\alpha)\mu\,(v_\alpha - v_2) = 0$$

and thus $v_\alpha = \alpha v_1 + (1-\alpha)v_2$. Finally,

$$\begin{aligned}\psi_\alpha &= \alpha\left[\psi_1 + \frac{\mu}{2}\|v_\alpha - v_1\|_2^2\right] + (1-\alpha)\left[\psi_2 + \frac{\mu}{2}\|v_\alpha - v_2\|_2^2\right] \\ &= \alpha\psi_1 + (1-\alpha)\psi_2 + \frac{\mu}{2}\left[\alpha(1-\alpha)^2\|v_2 - v_1\|_2^2 + (1-\alpha)\alpha^2\|v_2 - v_1\|_2^2\right] \\ &= \alpha\psi_1 + (1-\alpha)\psi_2 + \frac{\mu}{2}\alpha(1-\alpha)\|v_1 - v_2\|_2^2.\end{aligned}$$

□

B Regularized ERM duality

In this section we derive the dual (4) to the problem of computing proximal operator for the ERM objective (3) (Section B.1) and prove several bounds on primal and dual errors (Section B.2). Throughout this section we assume F is given by the ERM problem (1) and we make extensive use of the notation and assumptions in Section 1.1.

B.1 Dual derivation

We can rewrite the primal problem, $\min_x f_{s,\lambda}(x)$, as

$$\begin{array}{ll}\displaystyle\min_{x\in\mathbb{R}^d, z\in\mathbb{R}^n} & \sum_{i=1}^n \phi_i(z_i) + \frac{\lambda}{2}\|x - s\|_2^2 \\ \text{subject to} & z_i = a_i^\mathsf{T} x, \quad \text{for } i = 1,\ldots,n\end{array}.$$

By convex duality, this is equivalent to

$$\min_{x,\{z_i\}}\max_{y\in\mathbb{R}^n}\sum_{i=1}^n \phi_i(z_i) + \frac{\lambda}{2}\|x-s\|_2^2 + y^\mathsf{T}(Ax - z) = \max_y \min_{x,\{z_i\}}\sum_{i=1}^n \phi_i(z_i) + \frac{\lambda}{2}\|x-s\|_2^2 + y^\mathsf{T}(Ax - z)$$

Since

$$\min_{z_i}\{\phi_i(z_i) - y_i z_i\} = -\max_{z_i}\{y_i z_i - \phi_i(z_i)\} = -\phi_i^*(y_i)$$

and

$$\min_x\left\{\frac{\lambda}{2}\|x-s\|_2^2 + y^\mathsf{T}Ax\right\} = y^\mathsf{T}As + \min_x\left\{\frac{\lambda}{2}\|x-s\|_2^2 + y^\mathsf{T}A(x-s)\right\} = y^\mathsf{T}As - \frac{1}{2\lambda}\|A^\mathsf{T}y\|_2^2,$$

it follows that the optimization problem is in turn equivalent to

$$-\min_y \sum_{i=1}^n \phi_i^*(y_i) + \frac{1}{2\lambda}\|A^\mathsf{T}y\|_2^2 - s^\mathsf{T}A^\mathsf{T}y.$$

This negated problem is precisely the dual formulation.

The first problem is a Lagrangian saddle-point problem, where the Lagrangian is defined as

$$\mathcal{L}(x,y,z) = \sum_{i=1}^n \phi_i(z_i) + \frac{\lambda}{2}\|x-s\|_2^2 + y^\mathsf{T}(Ax - z).$$

The dual-to-primal mapping (5) and primal-to-dual mapping (6) are implied by the KKT conditions under $\mathcal{L}$, and can be derived by solving for x, y, and z in the system $\nabla\mathcal{L}(x, y, z) = 0$.

The *duality gap* in this context is defined as

$$\mathrm{gap}_{s,\lambda}(x, y) \overset{\mathrm{def}}{=} f_{s,\lambda}(x) + g_{s,\lambda}(y). \tag{20}$$

Strong duality dictates that $\mathrm{gap}_{s,\lambda}(x, y) \geq 0$ for all $x \in \mathbb{R}^d$, $y \in \mathbb{R}^n$, with equality attained when x is primal-optimal and y is dual-optimal.

B.2 Error bounds

Lemma B.1 (Dual error bounds primal error)**.** *For all $s \in \mathbb{R}^d$, $y \in \mathbb{R}^n$, and $\lambda > 0$ we have*

$$f_{s,\lambda}(\widehat{x}_{s,\lambda}(y)) - f^{opt}_{s,\lambda} \leq 2(n\kappa_\lambda)^2(g_{s,\lambda}(y) - g^{opt}_{s,\lambda}).$$

Proof. Because F is nR^2L smooth, $f_{s,\lambda}$ is $nR^2L + \lambda$ smooth. Consequently, for all $x \in \mathbb{R}^d$ we have

$$f_{s,\lambda}(x) - f^{\mathrm{opt}}_{s,\lambda} \leq \frac{nR^2L + \lambda}{2}\|x - x^{\mathrm{opt}}_{s,\lambda}\|_2^2$$

Since we know that $x^{\mathrm{opt}}_{s,\lambda} = s - \frac{1}{\lambda}A^\mathsf{T} y^{\mathrm{opt}}_{s,\lambda}$ and $\|A^\mathsf{T} z\|_2^2 \leq nR^2\|z\|_2^2$ for all $z \in \mathbb{R}^n$ we have

$$\begin{aligned} f_{s,\lambda}(\widehat{x}_{x,\lambda}(y)) - f_{s,\lambda}(x^{\mathrm{opt}}_{s,\lambda}) &\leq \frac{nR^2L + \lambda}{2}\|s - \frac{1}{\lambda}A^\mathsf{T} y - (s - \frac{1}{\lambda}A^\mathsf{T} y^{\mathrm{opt}}_{s,\lambda})\|_2^2 \\ &= \frac{nR^2L + \lambda}{2\lambda^2}\|y - y^{\mathrm{opt}}_{s,\lambda}\|^2_{AA^\mathsf{T}} \\ &\leq \frac{nR^2(nR^2L + \lambda)}{2\lambda^2}\|y - y^{\mathrm{opt}}_{s,\lambda}\|_2^2. \end{aligned} \tag{21}$$

Finally, since each ϕ_i^* is $1/L$-strongly convex, G is $1/L$-strongly convex and hence so is $g_{s,\lambda}$. Therefore by Lemma A.1 we have

$$\frac{1}{2L}\|y - y^{\mathrm{opt}}_{s,\lambda}\|_2^2 \leq g_{s,\lambda}(y) - g_{s,\lambda}(y^{\mathrm{opt}}_{s,\lambda}). \tag{22}$$

Substituting (22) in (21) and recalling that $\kappa_\lambda \geq 1$ yields the result. □

Lemma B.2 (Gap for primal-dual pairs)**.** *For all $s, x \in \mathbb{R}^d$ and $\lambda > 0$ we have*

$$\mathrm{gap}_{s,\lambda}(x, \widehat{y}(x)) = \frac{1}{2\lambda}\|\nabla F(x)\|_2^2 + \frac{\lambda}{2}\|x - s\|_2^2. \tag{23}$$

Proof. To prove the first identity (23), let $\widehat{y} = \widehat{y}(x)$ for brevity. Recall that

$$\widehat{y}_i = \phi_i'(a_i^\mathsf{T} x) \in \operatorname*{argmax}_{y_i}\{x^\mathsf{T} a_i y_i - \phi_i^*(y_i)\} \tag{24}$$

by definition, and hence $x^\mathsf{T} a_i\widehat{y}_i - \phi_i^*(\widehat{y}_i) = \phi_i(a_i^\mathsf{T} x)$. Observe that

$$\mathrm{gap}_{s,\lambda}(x, \widehat{y}) = \sum_{i=1}^n \Big(\phi_i(a_i^\mathsf{T} x) + \phi_i^*(\widehat{y}_i)\Big) - x^\mathsf{T} A^\mathsf{T}\widehat{y} + \tfrac{1}{2\lambda}\|A^\mathsf{T}\widehat{y}\|^2 + \tfrac{\lambda}{2}\|x - s\|^2$$

$$
\begin{aligned}
&= \sum_{i=1}^{n} \left(\underbrace{\phi_i(a_i^\mathsf{T} x) + \phi_i^*(\widehat{y}_i) - x^\mathsf{T} a_i \widehat{y}_i}_{=0 \text{ (by (24))}} \right) + \tfrac{1}{2\lambda}\|A^\mathsf{T}\widehat{y}\|^2 + \tfrac{\lambda}{2}\|x-s\|^2 \\
&= \tfrac{1}{2\lambda}\|A^\mathsf{T}\widehat{y}\|^2 + \tfrac{\lambda}{2}\|x-s\|^2 \\
&= \tfrac{1}{2\lambda}\|\sum_{i=1}^{n} a_i \phi_i'(a_i^\mathsf{T} x)\|^2 + \tfrac{\lambda}{2}\|x-s\|^2 \\
&= \tfrac{1}{2\lambda}\|\nabla F(x)\|^2 + \tfrac{\lambda}{2}\|x-s\|^2.
\end{aligned}
$$

□

Corollary B.3 (Initial dual error)**.** *For all $s, x \in \mathbb{R}^d$ and $\lambda > 0$ we have*

$$
g_{x,\lambda}(\widehat{y}(x)) - g_{x,\lambda}^{opt} \leq 2\kappa_\lambda \left(f_{x,\lambda}(x) - f_{x,\lambda}^{opt} \right)
$$

Proof. By Lemma B.2 we have

$$
\mathrm{gap}_{x,\lambda}(x, \widehat{y}(x)) = \frac{1}{2\lambda}\|\nabla F(x)\|_2^2 + \frac{\lambda}{2}\|x - x\|_2^2 = \frac{1}{2\lambda}\|\nabla F(x)\|_2^2
$$

Now clearly $\nabla F(x) = \nabla f_{x,\lambda}(x)$. Furthermore, since $f_{x,\lambda}(x)$ is $(nLR^2 + \lambda)$-smooth by Lemma A.1 we have $\|\nabla f_{x,\lambda}(x)\| \leq 2(nLR^2 + \lambda)(f_{x,\lambda}(x) - f_{x,\lambda}^{\mathrm{opt}})$. Consequently,

$$
g_{x,\lambda}(\widehat{y}(x)) - g_{x,\lambda}^{\mathrm{opt}} \leq \mathrm{gap}_{x,\lambda}(x, \widehat{y}(x)) \leq \frac{2(nLR^2 + \lambda)}{2\lambda} \left(f_{x,\lambda}(x) - f_{x,\lambda}^{\mathrm{opt}} \right) \quad .
$$

Recalling the definition of κ_λ and the fact that $1 \leq \kappa_\lambda$ yields the result. □