Designing Statistical Estimators That Balance Sample Size, Risk, and Computational Cost

John J. Bruer, Joel A. Tropp, Volkan Cevher, and Stephen R. Becker

***Abstract*—This paper proposes a tradeoff between computational time, sample complexity, and statistical accuracy that applies to statistical estimators based on convex optimization. When we have a large amount of data, we can exploit excess samples to decrease statistical risk, to decrease computational cost, or to trade off between the two. We propose to achieve this tradeoff by varying the amount of smoothing applied to the optimization problem. This work uses regularized linear regression as a case study to argue for the existence of this tradeoff both theoretically and experimentally. We also apply our method to describe a tradeoff in an image interpolation problem.**

***Index Terms*—Smoothing methods, statistical estimation, convex optimization, regularized regression, image interpolation, resource tradeoffs**

I. Motivation

MASSIVE DATA presents an obvious challenge to statistical algorithms. We expect that the computational effort needed to process a data set increases with its size. The amount of computational power available, however, is growing slowly relative to sample sizes. As a consequence, large-scale problems of practical interest require increasingly more time to solve. This creates a demand for new algorithms that offer better performance when presented with large data sets.

While it seems natural that larger problems require more effort to solve, Shalev-Shwartz and Srebro [1] showed that their algorithm for learning a support vector classifier actually becomes *faster* as the amount of training data increases. This and more recent works support an emerging viewpoint that treats data as a computational resource. That is, we should be able to exploit additional data to improve the performance of statistical algorithms.

We consider statistical problems solved through convex optimization and propose the following approach:

> *We can smooth statistical optimization problems more and more aggressively as the amount of available data increases. By controlling the amount of smoothing, we can exploit the additional data to decrease statistical risk, decrease computational cost, or trade off between the two.*

Our prior work [2] examined a similar time–data tradeoff achieved by applying a dual-smoothing method to (noiseless) regularized linear inverse problems. This paper generalizes those results, allowing for noisy measurements. The result is a tradeoff in computational time, sample size, and statistical accuracy.

We use regularized linear regression problems as a specific example to illustrate our principle. We provide theoretical and numerical evidence that supports the existence of a time–data tradeoff achievable through aggressive smoothing of convex optimization problems in the dual domain. Our realization of the tradeoff relies on recent work in convex geometry that allows for precise analysis of statistical risk. In particular, we recognize the work done by Amelunxen et al. [3] to identify phase transitions in regularized linear inverse problems and the extension to noisy problems by Oymak and Hassibi [4]. While we illustrate our smoothing approach using this single class of problems, we believe that many other examples exist.

A. Related Work and Our Contributions

Other researchers have identified related tradeoffs. Bottou and Bousquet [5] show that approximate optimization algorithms exhibit a tradeoff between small- and large-scale problems. Agarwal et al. [6] address a tradeoff between error and computational effort in statistical model selection problems. Shalev-Shwartz et al. [7] establish a time–data tradeoff in a binary classification problem. Berthet and Rigollet [8] provide rigorous lower bounds for sparse PCA that trade off computational efficiency and sample size. Daniely et al. [9] formally establish a time–data tradeoff in learning halfspaces over sparse vectors. Shender and Lafferty [10] identify a tradeoff by introducing sparsity into the covariance matrices of ridge regression problems. See [11] for a review of some recent perspectives on computational scalability that lead to time–data tradeoffs. Our work identifies a distinctly different tradeoff than these prior works.

Our approach bears most similarity to that of Chandrasekaran and Jordan [12]. They use an algebraic hierarchy of convex relaxations to achieve a time–data tradeoff for a class of denoising problems. The geometric intuition they develop also motivates our current work. In contrast, we use a continuous

The work of J. J. Bruer and J. A. Tropp was supported under ONR award N00014-11-1002, AFOSR award FA9550-09-1-0643, and a Sloan Research Fellowship. The work of V. Cevher was supported in part by the European Commission under the grants MIRG-268398 and ERC Future Proof, and by the Swiss Science Foundation under the grants SNF 200021-132548, SNF 200021-146750 and SNF CRSII2-147633. J. J. Bruer and J. A. Tropp are with the Department of Computing + Mathematical Sciences, California Institute of Technology, Pasadena, CA 91125 USA (email: jbruer@cms.caltech.edu; jtropp@cms.caltech.edu). V. Cevher is with the Laboratory for Information and Inference Systems, Swiss Federal Institute of Technology in Lausanne (EPFL), Lausanne, Switzerland (email: volkan.cevher@epfl.ch). S. R. Becker was with IBM Research, Yorktown Heights, NY 10598. He is now with the Department of Applied Mathematics, University of Colorado at Boulder, Boulder, CO 80309 USA (email: stephen.becker@colorado.edu).

sequence of relaxations based on smoothing and provide practical examples that are different in nature.

B. Roadmap to a Time–Data Tradeoff

In Section II, we present the regularized linear regression model. In Section III, we highlight recent work that establishes a geometric opportunity for a time–data tradeoff. In Section IV, we discuss the role of smoothing in solving convex optimization problems and describe a computational opportunity for a time–data tradeoff. We use a dual-smoothing scheme in Section V to seize both opportunities and create a time–data tradeoff. In Sections VI and VII we provide theoretical and numerical evidence of this tradeoff for sparse vector and low-rank matrix regression problems. We then use our approach to achieve a time–data tradeoff for an image interpolation problem in Section VIII.

II. Regularized Linear Regression

In this section, we describe the regularized linear regression problem that we use as a case study to illustrate our time–data tradeoff.

A. The Data Model

Assume that we have a data set $\{(\boldsymbol{a}_i, b_i) : i = 1, \dots, m\}$ comprising m *samples*, where the $\boldsymbol{a}_i \in \mathbb{R}^d$ are the *inputs*, and the $b_i \in \mathbb{R}$ are the *responses* of a statistical model. We call m the *sample size*, and we consider the case where $m < d$. Given a vector of parameters $\boldsymbol{x}^\natural \in \mathbb{R}^d$, we relate the inputs and responses through the linear equation

$$\boldsymbol{b} = \boldsymbol{A}\boldsymbol{x}^\natural + \boldsymbol{v}, \tag{1}$$

where the ith row of $\boldsymbol{A} \in \mathbb{R}^{m \times d}$ is the input $\boldsymbol{a}_i$, the ith entry of $\boldsymbol{b} \in \mathbb{R}^m$ is b_i, and the entries of $\boldsymbol{v} \in \mathbb{R}^m$ are independent, zero-mean random variates. The goal of the regression problem is to infer the underlying parameters $\boldsymbol{x}^\natural$ from the data.

B. Prediction Error and Statistical Risk

Let $\widehat{\boldsymbol{x}}$ be an estimate of the true vector $\boldsymbol{x}^\natural$. We evaluate the accuracy of this estimate using the notion of prediction error. The *average squared prediction error* of an estimate $\widehat{\boldsymbol{x}}$ is

$$R(\widehat{\boldsymbol{x}}) = \frac{1}{m} \|\boldsymbol{A}\widehat{\boldsymbol{x}} - \boldsymbol{A}\boldsymbol{x}^\natural\|^2 . \tag{2}$$

For a given measurement matrix $\boldsymbol{A}$ and parameter vector $\boldsymbol{x}^\natural$ in the data model (1), we call $\mathbb{E}_{\boldsymbol{v}}[R(\widehat{\boldsymbol{x}})]$ the *statistical risk* of the estimator.

Without knowing the true parameters $\boldsymbol{x}^\natural$, we cannot compute this quantity. We can, however, measure how closely the estimate $\widehat{\boldsymbol{x}}$ relates the inputs to the (noisy) observations in our given data set by computing

$$\widehat{R}(\widehat{\boldsymbol{x}}) := \frac{1}{m} \|\boldsymbol{A}\widehat{\boldsymbol{x}} - \boldsymbol{b}\|^2 . \tag{3}$$

The quantity $\widehat{R}(\widehat{\boldsymbol{x}})$ is an estimate of $R(\widehat{\boldsymbol{x}})$. In the regression setting, this is the (normalized) *residual sum of squares*; we will call it the *empirical risk*.

C. The Regularized Linear Regression Problem

In linear regression, it is common to require that $\widehat{\boldsymbol{x}}$ minimize the empirical risk. In our case, however, we have fewer samples than the number of parameters (i.e., $m < d$), and so we instead solve

$$\widehat{\boldsymbol{x}} := \arg\min_{\boldsymbol{x}} f(\boldsymbol{x}) \quad \text{subject to} \quad \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\| \le \sqrt{m \cdot R_{\max}} =: \epsilon, \tag{4}$$

where the proper convex function $f : \mathbb{R}^d \to \mathbb{R} \cup \{+\infty\}$ is a *regularizer*, and $R_{\max}$ is the maximal empirical risk we will tolerate. The following sections illustrate the potential for a time–data tradeoff in solving this optimization problem.

III. A Geometric Opportunity

In this section, we discuss how sample size affects the robustness of the regularized regression problem (4) to noise. The connection leads to a geometric opportunity for a time–data tradeoff.

A. Descent Cones and Statistical Dimension

Before we can introduce the relevant result, we must provide two definitions.

Definition III.1 (Descent cone)**.** The *descent cone* of a proper convex function $f : \mathbb{R}^d \to \mathbb{R} \cup \{+\infty\}$ at the point $\boldsymbol{x} \in \mathbb{R}^d$ is the convex cone

$$\mathcal{D}(f; \boldsymbol{x}) := \bigcup_{\tau > 0} \left\{ \boldsymbol{y} \in \mathbb{R}^d : f(\boldsymbol{x} + \tau \boldsymbol{y}) \le f(\boldsymbol{x}) \right\}.$$

The descent cone $\mathcal{D}(f; \boldsymbol{x})$ comprises the directions that decrease f locally at $\boldsymbol{x}$. We quantify the "size" of this convex cone using the notion of *statistical dimension*.

Definition III.2 (Statistical dimension [3, Def. 2.1])**.** Let $C \in \mathbb{R}^d$ be a closed convex cone. Its *statistical dimension* $\delta(C)$ is defined as

$$\delta(C) := \mathbb{E}_{\boldsymbol{g}} \left[\|\boldsymbol{\Pi}_C(\boldsymbol{g})\|^2 \right],$$

where $\boldsymbol{g} \in \mathbb{R}^d$ has independent standard Gaussian entries, and $\boldsymbol{\Pi}_C$ is the projection operator onto C.

The quantity $\delta(\mathcal{D}(f; \boldsymbol{x}^\natural))$ plays a critical role in the behavior of the regularized linear regression problem (4).

B. A Phase Transition

Amelunxen et al. [3] proved that, under certain randomized data models with noiseless measurements, the regularized linear regression problem (4) undergoes a phase transition when the number m of samples equals $\delta(\mathcal{D}(f; \boldsymbol{x}^\natural))$. Oymak and Hassibi [4] characterized the stability of this phase transition in the presence of noise. Their work considers the formulation

$$\text{minimize} \quad \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|^2 \quad \text{subject to} \quad f(\boldsymbol{x}) \le f(\boldsymbol{x}^\natural), \tag{5}$$

which is equivalent to (4) for some choice of the parameter $R_{\max}$. We present a restatement of their result here using our notational conventions.

Fact III.3 (The phase transition for regularized linear regression [4, Thm. 3.3])**.** *Assume that the measurement matrix $\boldsymbol{A}$ is chosen according to the Haar measure on the ensemble of matrices in $\mathbb{R}^{m\times d}$ with orthonormal rows. In particular, this requires $m \le d$. For such a measurement matrix, let $\boldsymbol{b} = \boldsymbol{A}\boldsymbol{x}^\natural + \boldsymbol{v}$ be a (random) observation vector with $\boldsymbol{v} \sim \text{NORMAL}(\boldsymbol{0}, \sigma^2 \mathbf{I}_m)$.*

Let $\boldsymbol{x}^\star$ (which depends on $\boldsymbol{b}$ and $\boldsymbol{A}$) be a minimizer of (5)*. Set $\delta = \delta(\mathcal{D}(f; \boldsymbol{x}^\natural))$. Let $R(\boldsymbol{x}^\star)$ denote the average squared prediction error* (2) *and $\widehat{R}(\boldsymbol{x}^\star)$ denote the empirical risk* (3) *of $\boldsymbol{x}^\star$. Then there exist constants $c_1, c_2 > 0$ such that*

- *Whenever $m < \delta$,*

$$\max_{\sigma>0} \frac{\mathbb{E}_{\boldsymbol{v}}\left[R(\boldsymbol{x}^\star) \mid \boldsymbol{A}\right]}{\sigma^2} = 1,$$

and

$$\lim_{\sigma\to 0} \frac{\mathbb{E}_{\boldsymbol{v}}\left[\widehat{R}(\boldsymbol{x}^\star) \mid \boldsymbol{A}\right]}{\sigma^2} = 0,$$

with probability $1 - c_1 \exp\left(-c_2(m-\delta)^2/d\right)$.

- *Whenever $m > \delta$,*

$$\left|\max_{\sigma>0} \frac{\mathbb{E}_{\boldsymbol{v}}\left[R(\boldsymbol{x}^\star) \mid \boldsymbol{A}\right]}{\sigma^2} - \frac{\delta}{m}\right| \le t m^{-1}\sqrt{d},$$

and

$$\left|\lim_{\sigma\to 0} \frac{\mathbb{E}_{\boldsymbol{v}}\left[\widehat{R}(\boldsymbol{x}^\star) \mid \boldsymbol{A}\right]}{\sigma^2} - \left(1 - \frac{\delta}{m}\right)\right| \le t m^{-1}\sqrt{d},$$

with probability $1 - c_1 \exp(-c_2 t^2)$.

The probabilities are taken over $\boldsymbol{A}$.

Notice that this result indeed describes a phase transition at $m = \delta(\mathcal{D}(f; \boldsymbol{x}^\natural))$. When the number m of samples is smaller than this quantity, the worst-case statistical risk is simply the noise power σ^2, and the regularized linear regression problem has no robustness to noise. That is, as the number of samples increases towards the phase transition, the statistical accuracy of the solution does not improve. After crossing the phase transition, however, additional samples decrease the worst-case risk at the rate $1/m$.

Additionally, the result gives us guidance in choosing the parameter $R_{\max}$ in our formulation of the regularized linear regression problem (4). If we have a reasonable estimate of the noise power σ^2, we use the worst-case expected empirical risk and set

$$R_{\max} = \sigma^2\left(1 - \frac{\delta\left(\mathcal{D}(f; \boldsymbol{x}^\natural)\right)}{m}\right),$$

and therefore we set

$$\epsilon = \sigma\left(m - \delta\left(\mathcal{D}(f; \boldsymbol{x}^\natural)\right)\right)^{1/2}. \tag{6}$$

Remark III.4. Fact III.3 considers partial unitary measurement matrices $\boldsymbol{A}$. Oymak and Hassibi also present numerical experiments that exhibit similar behavior when $\boldsymbol{A}$ has independent standard Gaussian entries. While the location of the phase transition remains the same, the choice of the parameter $R_{\max}$ in the regression problem (4) then depends on the spectrum of $\boldsymbol{A}$.

C. A Geometric Opportunity

Chandrasekaran and Jordan [12] argue that enlarging convex constraint sets can make corresponding statistical optimization problems easier to solve. These geometric deformations, however, create a loss of statistical accuracy. In the presence of large amounts of data, they argue that one could tune the relaxation to trade off between statistical and computational performance.

We see a similar opportunity in the regularized linear regression problem and illustrate it in Fig. 1. By enlarging the sublevel sets of the regularizer f, we increase the statistical dimension of the descent cone of f at $\boldsymbol{x}^\natural$. Fact III.3 tells us that the solution to the regression problem with the relaxed regularizer $\tilde{f}$ will have higher risk. If, however, the relaxed regularizer results in a problem that is easier to solve computationally, then we have a tradeoff between sample size, computational time, and statistical accuracy.

While our work builds on the geometric motivation of [12], we employ an entirely different approach to realize the tradeoff. They use a discrete sequence of relaxations based on an algebraic hierarchy, while we propose a continuous sequence of relaxations based on a dual-smoothing technique.

IV. A Computational Opportunity

In this section, we discuss the computational benefit of smoothing optimization problems, and we show how smoothing in the dual domain can reduce the computational cost of solving the regularized linear regression problem.

A. Convexity and Smoothness

Let us start with two definitions we will need throughout the remainder of this section. We measure the convexity of a function f_μ using the notion of *strong convexity*.

Definition IV.1 (Strong convexity)**.** A function $f_\mu : \mathbb{R}^d \to \mathbb{R}$ is *μ-strongly convex* if there exists a positive constant μ such that the function

$$\boldsymbol{x} \mapsto f_\mu(\boldsymbol{x}) - \frac{\mu}{2}\|\boldsymbol{x}\|^2,$$

is convex.

Higher values of the constant μ correspond to "more convex" functions.

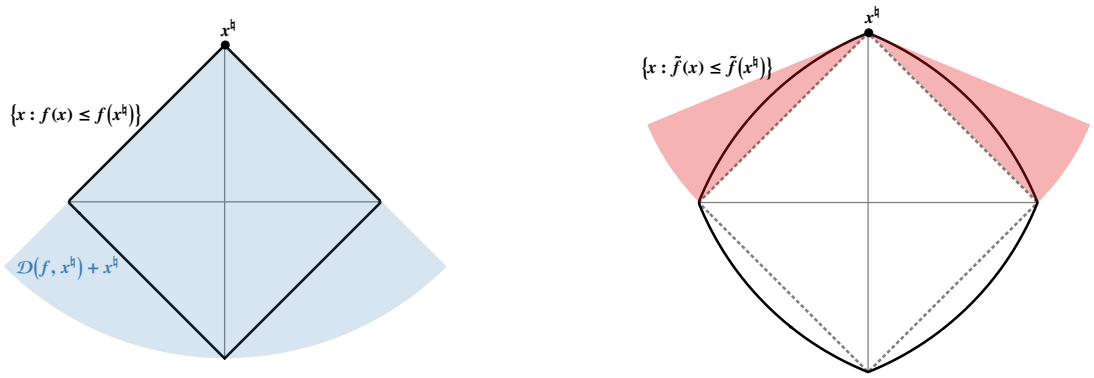
We measure the smoothness of a function g using the Lipschitz constant of its gradient ∇g.

Definition IV.2 (Lipschitz gradient)**.** A function $g : \mathbb{R}^m \to \mathbb{R}$ has an *L-Lipschitz gradient* if there exists a positive constant L such that

$$\|\nabla g(z_1) - \nabla g(z_2)\| \le L\|z_1 - z_2\|,$$

for all vectors $z_1, z_2 \in \mathbb{R}^m$.

Lower values of the Lipschitz constant L correspond to smoother functions g.



(a) (b)

Fig. 1. **A geometric opportunity.** Panel (a) illustrates the sublevel set and descent cone of a regularizer f at the point $\boldsymbol{x}^\natural$. Panel (b) shows a relaxed regularizer $\tilde{f}$ with larger sublevel sets. The shaded area indicates the difference between the descent cones of $\tilde{f}$ and f at $\boldsymbol{x}^\natural$. Fact III.3 shows how this difference in the size of the descent cones translates into a difference in statistical accuracy. We may compensate for this loss of statistical accuracy by choosing a relaxation $\tilde{f}$ that allows us to solve the optimization problem faster.

B. The Benefit of Smoothness

We focus on first-order methods—iterative algorithms that only require knowledge of the objective value and a gradient (or subgradient) at any given point—to solve the regularized linear regression problem (4). Nemirovski and Yudin [13] show that the best achievable convergence rate for such an algorithm that minimizes a convex objective with a Lipschitz gradient is $O(1/\gamma^{1/2})$ iterations, where γ is the numerical accuracy. Nesterov [14] provides an algorithm achieving this rate, the first of a class of algorithms known as accelerated gradient methods. For a unified framework describing these algorithms and their convergence properties, see [15].

Common choices of regularizer in (4), such as the ℓ_1 norm, are nonsmooth. Nesterov [16] provides a method to approximate some nonsmooth objectives with smooth ones. He shows that a specific class of first-order methods can then solve the smoothed problem at a faster convergence rate, albeit with some approximation error. Beck and Teboulle [17] generalize Nesterov's approach.

While applying a primal smoothing method to the regularized linear regression problem (4) may seem attractive, the geometric opportunity described in the previous section relies critically on the nonsmoothness of the regularizer f. Indeed, if we smooth f by any amount, its descent cones become halfspaces, and we lose all control over their size. We instead consider a method to smooth the dual of the optimization problem. This technique preserves the geometric opportunity while allowing for a computational speedup.

C. The Dual Problem

The properties of smoothness and convexity relate to each other through duality. We present a variation of a result in [18].

Fact IV.3 (The duality between convexity and smoothing [18, Prop. 12.60])**.** *If the proper closed convex function $f_\mu : \mathbb{R}^d \to \mathbb{R} \cup \{+\infty\}$ is μ-strongly convex, then its convex conjugate $f_\mu^* : \mathbb{R}^d \to \mathbb{R} \cup \{+\infty\}$ is differentiable and ∇f_μ^* is $\frac{1}{\mu}$-Lipschitz, where $f_\mu^*(\boldsymbol{x}^*) = -\inf_{\boldsymbol{x} \in \mathbb{R}^d} \left\{ f_\mu(\boldsymbol{x}) - \langle \boldsymbol{x}^*,\ \boldsymbol{x} \rangle \right\}$.*

As the convexity of the function f_μ increases, so does the smoothness of its conjugate f_μ^*. In order to see how we may exploit this duality between convexity and smoothing, we must first derive the Lagrangian dual of the regularized linear regression problem.

We replace the regularizer f in the regularized linear regression problem (4) with a μ-strongly convex function f_μ to obtain new estimators of the form

$$\widehat{\boldsymbol{x}}_\mu := \arg\min_{\boldsymbol{x}} f_\mu(\boldsymbol{x}) \quad \text{subject to} \quad \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\| \le \epsilon. \tag{7}$$

The dual problem is then

$$\begin{aligned} \text{maximize} \quad & g_\mu(z, t) := \inf_{\boldsymbol{x}} \left\{ f_\mu(\boldsymbol{x}) - \begin{pmatrix} z \\ t \end{pmatrix}^T \begin{pmatrix} \boldsymbol{A}\boldsymbol{x} - \boldsymbol{b} \\ \epsilon \end{pmatrix} \right\} \\ \text{subject to} \quad & \|z\| \le t, \end{aligned}$$

where we used the fact that $\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\| \le \epsilon$ is a conic constraint, and the second-order cone is self-dual. Since $\epsilon \ge 0$, we can eliminate the dual variable t to obtain the unconstrained problem

$$\text{maximize} \quad g_\mu(z) := \inf_{\boldsymbol{x}} \left\{ f_\mu(\boldsymbol{x}) - \langle z,\ \boldsymbol{A}\boldsymbol{x} - \boldsymbol{b} \rangle - \epsilon \|z\| \right\}. \tag{8}$$

Note that the dual function g_μ is not smooth. We can, however, rewrite it as the composite function

$$\begin{aligned} g_\mu(z) &= \inf_{\boldsymbol{x}} \left\{ f_\mu(\boldsymbol{x}) - \langle \boldsymbol{A}^T z,\ \boldsymbol{x} \rangle \right\} + \langle z,\ \boldsymbol{b} \rangle - \epsilon \|z\| \\ &= \underbrace{-f_\mu^*(\boldsymbol{A}^T z) + \langle z,\ \boldsymbol{b} \rangle}_{\tilde{g}_\mu(z)} \underbrace{- \epsilon \|z\|}_{h(z)}, \end{aligned} \tag{9}$$

where f_μ^* is the convex conjugate of f_μ. We use the strong convexity of the regularizer f_μ to show that $\tilde{g}_\mu$ has a Lipschitz gradient, and so this is really a decomposition of the dual function g_μ into smooth ($\tilde{g}_\mu$) and nonsmooth (h) components. In particular, we have the following lemma.

Lemma IV.4. *Let $f_\mu : \mathbb{R}^d \to \mathbb{R} \cup \{+\infty\}$ be the regularizer in the regression problem* (7)*. Assume that f_μ is coercive (i.e.,*

Algorithm 1. **Auslender–Teboulle**

Input: measurement matrix $\boldsymbol{A}$, observed vector $\boldsymbol{b}$, parameter ϵ

1: $z_0 \leftarrow \mathbf{0}$, $\bar{z}_0 \leftarrow z_0$, $\theta_0 \leftarrow 1$
2: **for** $k = 0, 1, 2, \ldots$ **do**
3: $\quad \boldsymbol{y}_k \leftarrow (1-\theta_k)z_k + \theta_k \bar{z}_k$
4: $\quad \boldsymbol{x}_k \leftarrow \arg\min_{\boldsymbol{x}} \{f(\boldsymbol{x}) + \langle \boldsymbol{y}_k,\ \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}\rangle\}$
5: $\quad \bar{z}_{k+1} \leftarrow \text{Shrink}\left(\bar{z}_k - (\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_k)/(L_\mu \cdot \theta), \epsilon/(L_\mu \cdot \theta)\right)$
6: $\quad z_{k+1} \leftarrow (1-\theta_k)z_k + \theta_k \bar{z}_{k+1}$
7: $\quad \theta_{k+1} \leftarrow 2/(1 + (1 + 4/\theta_k^2)^{1/2})$
8: **end for**

$f_\mu(\boldsymbol{x}) \to +\infty$ as $\|\boldsymbol{x}\| \to +\infty$) and μ-strongly convex. Then the function $\tilde{g}_\mu$ as in (9) *has gradient*

$$\nabla \tilde{g}_\mu(z) = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_z,$$

where

$$\boldsymbol{x}_z := \arg\min_{\boldsymbol{x}} \left\{ f_\mu(\boldsymbol{x}) - \langle \boldsymbol{A}^T z,\ \boldsymbol{x}\rangle \right\}. \tag{10}$$

Furthermore, $\nabla \tilde{g}_\mu$ is Lipschitz continuous with Lipschitz constant at most $L_\mu := \mu^{-1}\|\boldsymbol{A}\|^2$.

Proof: As given in (9), we have that

$$\tilde{g}_\mu(z) = \langle z,\ \boldsymbol{b}\rangle - f_\mu^*(\boldsymbol{A}^T z),$$

where f_μ^* is the convex conjugate of f_μ. Since we have assumed that f_μ is μ-strongly convex, Fact IV.3 tells us that f_μ^* is differentiable. Indeed,

$$f_\mu^*(\boldsymbol{A}^T z) = -\inf_{\boldsymbol{x}} \left\{ f_\mu(\boldsymbol{x}) - \langle \boldsymbol{A}^T z,\ \boldsymbol{x}\rangle \right\}.$$

The coercivity and strong convexity of f_μ guarantee both that the infimum is attained and that the minimizer is unique. Therefore, $\nabla f_\mu^*(\boldsymbol{A}^T z) = \boldsymbol{A}\boldsymbol{x}_z$, with $\boldsymbol{x}_z$ as given by (10).

Furthermore, Fact IV.3 tells us that ∇f_μ^* is Lipschitz continuous with parameter μ^{-1}. Therefore,

$$\begin{aligned}\|\nabla \tilde{g}_\mu(z_1) - \nabla \tilde{g}_\mu(z_2)\| &= \|\boldsymbol{A} \cdot \left(\nabla f_\mu^*(\boldsymbol{A}^T z_2) - \nabla f_\mu^*(\boldsymbol{A}^T z_1)\right)\| \\ &\le \|\boldsymbol{A}\| \cdot \mu^{-1} \|\boldsymbol{A}^T z_2 - \boldsymbol{A}^T z_1\| \\ &\le \mu^{-1} \|\boldsymbol{A}\|^2 \|z_2 - z_1\|,\end{aligned}$$

for all $z_1, z_2 \in \mathbb{R}^m$. ∎

We can now solve the composite dual problem (8) using an accelerated gradient method [19], [15]. Provided that the regularized regression problem (7) is strictly feasible, then strong duality holds for (7) and (8) by Slater's condition [20, Sec. 5.2.3]. Note that $\boldsymbol{A}$ having full row rank is sufficient to guarantee strict feasibility. Therefore, if we solve the dual problem (8) to obtain an optimal dual point $z^\star$, we may use (10) to find the unique optimal primal point $\boldsymbol{x}_{z^\star}$.

D. Example: Auslender–Teboulle

In Algorithm 1, we list an accelerated gradient method originally due to Auslender and Teboulle [21] and adapted by Becker et al. [22] to the regularized regression problem. We use this algorithm as an example to illustrate our time–data tradeoff, and the following analysis could be performed for other iterative methods. In particular, recent work by Tran-Dinh and Cevher [23] provides a first-order primal–dual framework that contains the necessary convergence guarantees on the primal feasibility gap.

Note that Algorithm 1, line 5 is the solution to the *composite gradient mapping* (see [19])

$$\bar{z}_{k+1} \leftarrow \arg\min_{z \in \mathbb{R}^m} \left\{ \tilde{g}_\mu(z_k) + \langle -\nabla \tilde{g}_\mu(z_k),\ z - z_k\rangle + \frac{1}{2} L_\mu \theta_k \|z - \bar{z}_k\|^2 + h(z) \right\},$$

and the map Shrink is given by

$$\text{Shrink}(z, t) = \max\left\{1 - \frac{t}{\|z\|}, 0\right\} \cdot z.$$

We have the following bound on the feasibility gap of primal iterates $\boldsymbol{x}_k$ at each iteration k.

Theorem IV.5 (Primal feasibility gap)**.** *Assume that the regularizer f_μ in the linear regression problem* (7) *is μ-strongly convex. Apply Algorithm 1 to the corresponding dual problem* (8)*, and let $z^\star$ be the optimal dual point. For any $k \ge 0$,*

$$\left| \|\boldsymbol{A}\boldsymbol{x}_k - \boldsymbol{b}\| - \epsilon \right| \le \frac{2\|\boldsymbol{A}\|^2 \|z^\star\|}{\mu k}. \tag{11}$$

See Appendix B for the proof. Note that the right-hand side of the bound (11) becomes smaller as the strong convexity parameter μ increases (or equivalently, as the Lipschitz constant L_μ decreases).

We can relate (11) back to the empirical risk (3) of $\boldsymbol{x}_k$ by recalling that $\|\boldsymbol{A}\boldsymbol{x}_k - \boldsymbol{b}\| = (m\widehat{R}(\boldsymbol{x}_k))^{1/2}$ and $\epsilon = (mR_{\max})^{1/2}$, where m is the sample size. Disregarding the impact of μ on the size of the optimal point $z^\star$, this bound suggests that, as the convexity of the regularizer f_μ increases, the number of iterations sufficient for Algorithm 1 to converge to the preset empirical risk target $R_{\max}$ decreases.

E. A Computational Opportunity

The geometric opportunity in Section III suggests replacing the regularizer f in the regularized linear regression problem (4) with a relaxed regularizer $\tilde{f}$ that is easier to optimize. Theorem IV.5 suggests that choosing f_μ in (7) to be a strongly convex approximation of f is a suitable relaxation.

Becker et al. [22] previously explored replacing non-strongly convex regularizers with strongly convex relaxations in order to achieve computational speedups in conic optimization problems (including the regularized linear regression problem). In their work, however, the amount of relaxation was chosen in an *ad hoc* manner primarily to facilitate the use of accelerated gradient methods. Instead, we propose to synthesize the above geometric and computational opportunities into a tunable time–data tradeoff, whereby we can choose the amount of relaxation in a principled manner.

V. A Time–Data Tradeoff

In this section, we show how to achieve a time–data tradeoff by exploiting both the geometric and computational opportunities of the previous sections.

A. A Dual-smoothing Method

In Section IV, we showed that if the regularizer in the regression problem (7) is strongly convex, then we may use an accelerated gradient method such as Algorithm 1 to solve the dual problem (8). Many common regularizers such as the ℓ_1 norm are not strongly convex, and so we must provide an appropriate relaxation method before applying Algorithm 1.

The procedure we use essentially applies Nesterov's primal-smoothing method from [16] to the dual problem; see [22]. Given a regularizer f in (4), we introduce a family $\{f_\mu : \mu > 0\}$ of strongly convex majorants:

$$f_\mu(\boldsymbol{x}) := f(\boldsymbol{x}) + \frac{\mu}{2} \|\boldsymbol{x}\|^2 .$$

Clearly, f_μ is μ-strongly convex, and so we may use any of these relaxations as the objective in (7). Note that these majorants also have larger sublevel sets, and their descent cones have larger statistical dimension. They are indeed relaxations that allow us to realize both the geometric and computational opportunities of the previous two sections.

B. Computational Cost

To assess the computational cost of solving (8), we must know two things: the number of iterations necessary for convergence and the cost of each iteration. In practice, we terminate Algorithm 1 when the relative primal feasibility gap $|\|\boldsymbol{A}\boldsymbol{x}_k - \boldsymbol{b}\| - \epsilon| / \epsilon$ is smaller than some tolerance. Theorem IV.5 allows us to bound the number of iterations sufficient to guarantee this occurrence.

Corollary V.1 (Iteration bound)**.** *Apply Algorithm 1 to solve the smoothed dual problem* (8)*, and let $\boldsymbol{z}^\star$ be the optimal dual point. Assume that the measurement matrix $\boldsymbol{A}$ has orthonormal rows and that the noise vector $\boldsymbol{v}$ in the data model* (1) *has distribution* NORMAL$(\boldsymbol{0}, \sigma^2 \mathbf{I}_m)$*, so that Fact III.3 applies. Set $\epsilon = \sigma(m - \delta)^{1/2}$, where $\delta = \delta(\mathcal{D}(f_\mu; \boldsymbol{x}^\natural))$; cf.* (6)*. Terminate the algorithm when the relative primal feasibility gap $|\|\boldsymbol{A}\boldsymbol{x}_k - \boldsymbol{b}\| - \epsilon| / \epsilon \le \gamma$. Then the number k of iterations sufficient for convergence satisfies the upper bound*

$$k \le \frac{2 \|\boldsymbol{z}^\star\|}{\gamma \mu \sigma \sqrt{m - \delta}}.$$

Note that increasing the smoothing parameter μ will also cause δ, and possibly $\|\boldsymbol{z}^\star\|$, to increase. This suggests some limit to the amount of dual-smoothing we may apply to the regularizer f for any given sample size m if we are to achieve a computational speedup.

The particular choice of the original regularizer f affects the cost of each iteration only in Algorithm 1, line 4. Fortunately, many regularizers of interest admit relatively inexpensive solutions to this subproblem by way of their proximity operators; we will see specific examples in the following sections. Provided that this step is indeed inexpensive, the dominant cost of each iteration comes from calculating the matrix–vector products involving the measurement matrix $\boldsymbol{A} \in \mathbb{R}^{m \times d}$. Therefore, the dominant cost of each iteration is $O(md)$.

In particular, this suggests that the computational cost will rise as $O(m^{1/2})$ if the smoothing parameter μ (and hence δ) stays constant as the sample size m increases. Increasing the smoothing parameter μ is critical for achieving a speedup when we have more samples.

C. Choosing a Smoothing Parameter

Choosing an appropriate value for the smoothing parameter μ is vital. The result due to Oymak and Hassibi [4]—given as Fact III.3—tells us both the number of samples required such that the estimator (7) is robust to noise and how the statistical error varies with the number of samples. We look at three schemes for choosing μ that satisfy different goals. Taken together, these schemes span the time–data tradeoff.

1) Constant Smoothing: The simplest method requires us to choose a constant value of μ. Larger values of μ lead to larger values of $\delta = \delta(\mathcal{D}(f_\mu; \boldsymbol{x}^\natural))$, the location of the phase transition. Additionally, they lead to higher worst-case levels of statistical risk. Therefore, we choose a relatively small value of μ, minimizing the error introduced by the relaxation of the regularizer. Let us call this baseline value $\overline{\mu}$ and let $\overline{\delta} := \delta(\mathcal{D}(f_{\overline{\mu}}; \boldsymbol{x}^\natural))$. We will reference these quantities in the following schemes. Under this scheme, computational cost rises and statistical risk falls as the sample size increases.

Remark V.2. It is important to note that the phase transition occurs over a region of sample sizes. To avoid this area, we specify a baseline number of measurements $\overline{m}$ that is greater than the baseline statistical dimension $\overline{\delta}$. Choosing $\overline{m} = \overline{\delta} + \sqrt{d}$ appears sufficient and conservative. For a further discussion of the phase transition region, see [3] and the subsequent work [24] with refined results.

2) Constant Risk: If the number m of samples is greater than the baseline sample size $\overline{m}$, that means we could relax the regularizer further—by increasing μ—while maintaining the baseline level of risk. To do this, we choose the largest value of μ such that

$$\frac{\delta\left(\mathcal{D}(f_\mu; \boldsymbol{x}^\natural)\right)}{m} = \frac{\overline{\delta}}{\overline{m}}.$$

Note that this results in the lowest computational cost while retaining robustness to noise (at a fixed level of risk).

3) A Tunable Balance: In reality, however, we will want some compromise between these two schemes. The constant smoothing scheme will become increasingly more expensive computationally, and the constant risk scheme provides no statistical improvement as the number of samples grows.

The idea behind this balanced scheme is to increase the smoothing parameter μ in a way such that both the computational cost *and* the risk decrease as the sample size increases. We choose a scaling parameter $\alpha \le 1$ and set μ to be the largest value such that

$$\frac{\delta\left(\mathcal{D}(f_\mu; \boldsymbol{x}^\natural)\right)}{m} = \frac{\overline{\delta}}{\overline{m} + (m - \overline{m})^\alpha}. \tag{12}$$

Recall that under the constant smoothing scheme, the risk will scale as m^{-1}, where $m \ge \overline{m}$ is the number of samples. Under the balanced scheme, however, we take the excess measurements $m - \overline{m}$ and have them effectively reduce the risk at the rate

$m^{-\alpha}$. Note that choosing $\alpha = 1$ recovers the constant smoothing scheme, while choosing $\alpha = -\infty$ recovers to the constant risk scheme. The "best" choice of α depends on the priorities of the practitioner.

D. Connection to the Noiseless Problem

Our previous work [2] examined the case where the linear measurements in the data model (1) contained no noise, i.e., when $\boldsymbol{v} = \mathbf{0}$. In that case, we used a heuristic choice of the smoothing parameter μ to effect a time–data tradeoff. The approach in this paper, however, lets us consider the noiseless problem as simply a special case of the noisy version.

Without noise in the measurements, we can recover the unknown signal $\boldsymbol{x}^\natural$ exactly. In other words, the recovered estimate has zero statistical risk. Therefore, our tradeoff in computational time, sample size, and statistical accuracy collapses to one in time and sample size only. As such, we may use all of the excess samples to reduce computational time, and we pay no penalty in statistical risk. By choosing the smoothing parameter using the "constant risk" method above, we recover the unknown signal faster than the "constant smoothing" method without losing any accuracy. Note that we recover the noiseless optimization problem and algorithm by choosing the maximum tolerated empirical risk $R_{\max} = 0$ in regularized regression problem (7).

E. The Time–Data Tradeoff

We summarize the tradeoff between time, data, and accuracy as follows:

> *When we have excess samples in the data set, we can exploit them to decrease the statistical risk of our estimator or to lower the computational cost through additional smoothing. A tradeoff arises from the balance between these two competing interests.*

The following sections evidence the existence of this tradeoff for particular examples. We emphasize, however, that the main idea of combining these geometric and computational opportunities to realize a tradeoff is more broadly applicable.

VI. Example: Sparse Vector Regression

In this section, we examine a time–data tradeoff for sparse vector regression problems.

A. The Dual-smoothed Problem

Assume that the parameter vector $\boldsymbol{x}^\natural \in \mathbb{R}^d$ in the data model (1) is sparse. The ℓ_1 norm serves as a convex proxy for sparsity, so we choose it as the regularizer in the regression problem (4). This problem is equivalent to the LASSO of Tibshirani [25].

We apply the dual-smoothing procedure from Section V-A to obtain the relaxed regularizer

$$f_\mu(\boldsymbol{x}) = \|\boldsymbol{x}\|_{\ell_1} + \frac{\mu}{2}\|\boldsymbol{x}\|^2. \tag{13}$$

The corresponding primal problem (7) is equivalent to the elastic net of Zou and Hastie [26]. The composite dual is given by (9).

To apply Algorithm 1 to the dual-smoothed sparse vector regression problem, we must calculate the primal iterate $\boldsymbol{x}_k$ from the current dual iterate $\boldsymbol{y}_k$ (Algorithm 1, line 4). This step can be written as

$$\boldsymbol{x}_k \leftarrow \mu \cdot \text{SoftThresh}(\boldsymbol{A}^T\boldsymbol{y}_k, 1),$$

where SoftThresh is the map given component-wise by

$$[\text{SoftThresh}(\boldsymbol{x}, t)]_i = \text{sgn}(x_i) \cdot \max\{|x_i| - t, 0\}.$$

This operation is inexpensive, and so the total cost of each iteration in Algorithm 1 is $O(md)$ operations.

B. Calculating the Statistical Dimension

In order to choose the smoothing parameter μ using Section V-C, we need to be able to calculate the statistical dimension of the descent cone of the relaxed regularizer f_μ at a sparse vector $\boldsymbol{x}$. The following result provides an upper bound on this quantity that depends only on the sparsity of $\boldsymbol{x}$ and the magnitude of its largest entry.

Proposition VI.1 (Statistical dimension bound for the dual-smoothed ℓ_1 norm)**.** *Let $\boldsymbol{x} \in \mathbb{R}^d$ be s-sparse, and define the normalized sparsity $\rho := s/d$. Let f_μ be as in* (13)*. Then*

$$\frac{\delta\left(\mathcal{D}(f_\mu; \boldsymbol{x})\right)}{d} \le \psi(\rho),$$

where $\psi\colon [0,1] \to \mathbb{R}$ is the function given by

$$\psi(\rho) = \inf_{\tau \ge 0}\left\{\rho\left[1 + \tau^2(1 + \mu\|\boldsymbol{x}\|_{\ell_\infty})^2\right] + (1-\rho)\sqrt{\frac{2}{\pi}}\int_\tau^\infty (u-\tau)^2 \mathrm{e}^{-u^2/2}\,\mathrm{d}u\right\}.$$

The proof is substantially similar to that in [3]. In our case, however, the resulting function ψ depends on the magnitudes of the nonzero entries of the vector $\boldsymbol{x}$. We use $\|\boldsymbol{x}\|_{\ell_\infty}$ as the upper bound for each entry in order to establish the proposition. Therefore, our result is most accurate for signals $\boldsymbol{x}$ that have low dynamic range (i.e., the nonzero entries of $\boldsymbol{x}$ have magnitude close to $\|\boldsymbol{x}\|_{\ell_\infty}$). Note that $\psi(0) = 0$, $\psi(1) = 1$, and ψ is increasing. Furthermore, as we increase the smoothing parameter μ, the statistical dimension will increase for $\rho \in (0, 1)$.

With this information, we can now examine the time–data tradeoff resulting from the smoothing schemes presented in Section V-C.

C. Numerical Experiment

In Fig. 2, we show the results of a numerical experiment that reveals the time–data tradeoff enabled by the smoothing schemes in Section V-C. See Appendix A-A for the methodological details.

Most practitioners use a fixed smoothing parameter μ that depends on the ambient dimension or sparsity but *not* on the sample size. For the constant smoothing case, we choose the smoothing parameter $\mu = 0.1$ based on the recommendation in [27] for the noiseless case. It is common, however, to see

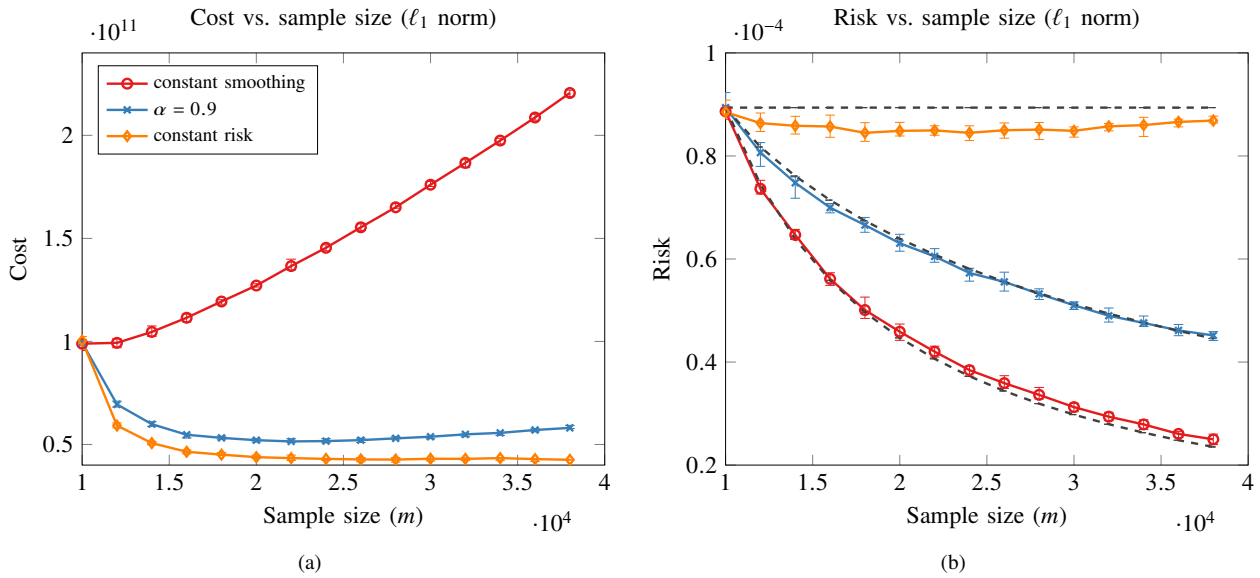


Fig. 2. **Sparse vector regression experiment.** The panels show (a) the average computational cost and (b) the estimated statistical risk over 10 random trials of the dual-smoothed sparse vector regression problem with ambient dimension $d = 40\,000$, normalized sparsity $\rho = 5\%$, and noise level $\sigma = 0.01$ for various sample sizes m. The red curve (circles) represents using a fixed smoothing parameter $\mu = 0.1$, the orange curve (diamonds) results from adjusting the smoothing parameter μ to maintain the baseline risk, and the blue curve (crosses) uses the balanced scheme (12) with scaling parameter $\alpha = 0.9$. For all schemes, the baseline smoothing parameter $\bar{\mu} = 0.1$, and the baseline sample size $\bar{m} = 10\,000$. The error bars indicate the minimum and maximum observed values. The dashed black lines show the predicted risk based on Proposition VI.1 and Fact III.3.

much smaller choices of μ; see [28], [29]. We compare this to the constant risk case and our balanced method with $\alpha = 0.9$. We choose the scaling parameter $\alpha = 0.9$ simply as a demonstration. This serves to illustrate how additional samples allow us the flexibility to trade off statistical accuracy and computational cost.

In the experiment, we fix both the ambient dimension $d = 40\,000$ and the normalized sparsity $\rho = 5\%$. To test each smoothing approach, we generate and solve 10 random sparse vector regression problems for each value of the sample size $m = 10\,000, 12\,000, 14\,000, \ldots, 38\,000$. Each problem comprises a random measurement matrix $\boldsymbol{A} \in \mathbb{R}^{m \times d}$ with orthonormal rows and a random sparse vector $\boldsymbol{x}^\natural$ whose 2000 nonzero entries are ± 1. Both are chosen uniformly at random from their respective sets. We use the baseline smoothing parameter $\bar{\mu} = 0.1$ and the baseline sample size $\bar{m} = 10\,000$, which is roughly $\delta(\mathcal{D}(f_{\bar{\mu}}; \boldsymbol{x}^\natural)) + \sqrt{40\,000}$. We stop Algorithm 1 when the relative primal feasibility gap $|\|\boldsymbol{A}\boldsymbol{x}_k - \boldsymbol{b}\| - \epsilon| / \epsilon$ is less than 10^{-3}, where ϵ is set according to (6). This condition allows us to accurately predict the risk of the resulting estimator $\widehat{\boldsymbol{x}}$ by using Fact III.3.

In Fig. 2(a), we see that the total computational cost[1] increases with sample size under the constant smoothing scheme. Meanwhile, the constant risk scheme displays a decrease in total cost as the sample size increases. The balanced scheme, however, shows an initial drop in cost before rising again. This shows the high cost of performing dense matrix multiplication at each iteration. The balanced scheme ($\alpha = 0.9$) smooths more aggressively than the constant scheme, and so it achieves an overall speedup. It, however, smooths less aggressively than the constant risk scheme, and so it (like the constant smoothing scheme) cannot overcome the high cost of the matrix multiplications as the sample size grows.

Even so, the cost required for 38 000 samples under the constant smoothing scheme is 3.8× higher than that of the balanced scheme. Furthermore, the cost of the balanced scheme with 38 000 samples is still less than the cost of the constant smoothing scheme with 10 000 samples.

In order to determine whether the cost is worth paying, we refer to Fig. 2(b) showing the risk as a function of sample size. The constant risk scheme behaves as expected, and the constant smoothing decreases risk the most as the sample size increases. The risk in the balanced scheme decreases by a factor of 2.0× as the sample size grows from 10 000 to 38 000 samples.

While the risk under the balanced scheme is 1.8× the risk under the constant smoothing scheme at a sample size of 38 000, it requires roughly 1/4 of the computational cost. Put another way, as the balanced scheme moves from 10 000 samples to 38 000 samples, risk decreases by a factor of 2.0× *and* computational cost decreases by 1.7×.

Note that the risk predictions (depicted by black dashed lines) resulting from Fact III.3 and the statistical dimension calculation in Proposition VI.1 are quite accurate. This means that given a fixed sample size and a target risk level, we can actually calculate the necessary value of the smoothing parameter μ to achieve that risk level.

We emphasize that we use the same algorithm to test all three smoothing approaches, so the relative comparison between them is meaningful. The observed improvement shows that we have indeed identified a time–data tradeoff by smoothing.

[1]We compute total cost as $k \cdot md$, where k is the number of iterations taken, and md is the dominant cost of each iteration.

VII. Example: Low-rank Matrix Regression

In this section, we examine a time–data tradeoff for low-rank matrix regression problems.

A. The Dual-smoothed Problem

We may also use the data model (1) when the underlying signal is a matrix. Let $\boldsymbol{X}^\natural \in \mathbb{R}^{d_1 \times d_2}$ be the true matrix, and let $\boldsymbol{A} \in \mathbb{R}^{m \times d}$ be a measurement matrix, where $d := d_1 d_2$. Then the observations are given by $\boldsymbol{b} = \boldsymbol{A} \cdot \mathrm{vec}(\boldsymbol{X}^\natural)$, where vec returns the (column) vector obtained by stacking the columns of the input matrix.

Assume that $\boldsymbol{X}^\natural$ is low-rank. The Schatten 1-norm $\|\cdot\|_{S_1}$—the sum of the matrix's singular values—serves as a convex proxy for rank, and so we choose $f = \|\cdot\|_{S_1}$ as the regularizer in the regression problem (4). Toh and Yun consider this natural extension to the LASSO in [30]. We apply the dual-smoothing procedure from Section V-A to obtain the relaxed regularizer

$$f_\mu(\boldsymbol{X}) = \|\boldsymbol{X}\|_{S_1} + \frac{\mu}{2} \|\boldsymbol{X}\|_{\mathrm{F}}^2 . \tag{14}$$

The relaxed primal problem is again (7), and the composite dual is given by (9).

To apply Algorithm 1 to the dual-smoothed low-rank matrix regression problem, we must calculate the primal iterate $\boldsymbol{X}_k$ from the dual iterate $\boldsymbol{y}_k$ (Algorithm 1, line 4). This step can be written as

$$\boldsymbol{X}_k \leftarrow \mu \cdot \mathrm{SoftThreshSingVal}(\mathrm{mat}(\boldsymbol{A}^T \boldsymbol{y}_k), 1),$$

where SoftThreshSingVal applies soft-thresholding to the singular values of a matrix, and mat is the inverse of the vec operator. Given a matrix $\boldsymbol{X}$ and its SVD $\boldsymbol{U} \cdot \mathrm{diag}(\boldsymbol{\sigma}) \cdot \boldsymbol{V}^T$, we can express SoftThreshSingVal as

$$\mathrm{SoftThreshSingVal}(\boldsymbol{X}, t) = \boldsymbol{U} \cdot \mathrm{diag}\left(\mathrm{SoftThresh}(\boldsymbol{\sigma}, t)\right) \cdot \boldsymbol{V}^T,$$

where SoftThresh is simply the soft-thresholding operator on vectors. The SVD of $\mathrm{mat}(\boldsymbol{A}^T \boldsymbol{z})$ has cost $O(d_1 d_2^2) = O(d d_2)$ (for $d_1 \leq d_2$). Since the number m of measurements will be larger than d_2, the dominant cost of each iteration of Algorithm 1 is still $O(md)$ operations.

B. Calculating the Statistical Dimension

As in the sparse vector case, we must be able to compute the statistical dimension of the descent cones of f_μ at a given low-rank matrix $\boldsymbol{X}$. In the case where the unknown matrix is square, the following result gives an upper bound on the statistical dimension depending on the rank of the matrix and the magnitude of its largest singular value.

Proposition VII.1 (Statistical dimension bound for the dual-smoothed Schatten 1-norm)**.** *Let $\boldsymbol{X} \in \mathbb{R}^{d_1 \times d_1}$ have rank r, and define the normalized rank $\rho := r/d_1$. Let f_μ be as in* (14)*. Then*

$$\frac{\delta\left(\mathcal{D}(f_\mu; \boldsymbol{X})\right)}{d_1^2} \leq \psi(\rho) + \mathrm{o}(1),$$

where $\psi : [0, 1] \to \mathbb{R}$ is the function given by

$$\psi(\rho) := \inf_{0 \leq \tau \leq 2} \left\{ \rho + (1 - \rho) \left[\rho \left(1 + \tau^2 (1 + \mu \|\boldsymbol{X}\|)^2\right) + \frac{(1 - \rho)}{12\pi} \left[24(1 + \tau^2) \cos^{-1}(\tau/2) - \tau(26 + \tau^2)\sqrt{4 - \tau^2} \right] \right] \right\}.$$

The proof is substantially similar to that in [3]. Their technique also provides a statistical dimension bound when the matrix is non-square. In our case, the resulting function ψ depends on the magnitude of each of the nonzero singular values of $\boldsymbol{X}$. To establish our proposition, we use the largest singular value $\|\boldsymbol{X}\|$ as an upper bound. Therefore, this result is most accurate when all the nonzero singular values are close to $\|\boldsymbol{X}\|$. The behavior of ψ is also similar to that of the sparse vector example.

With this information, we can now examine the time–data tradeoff resulting from the smoothing schemes presented in Section V-C.

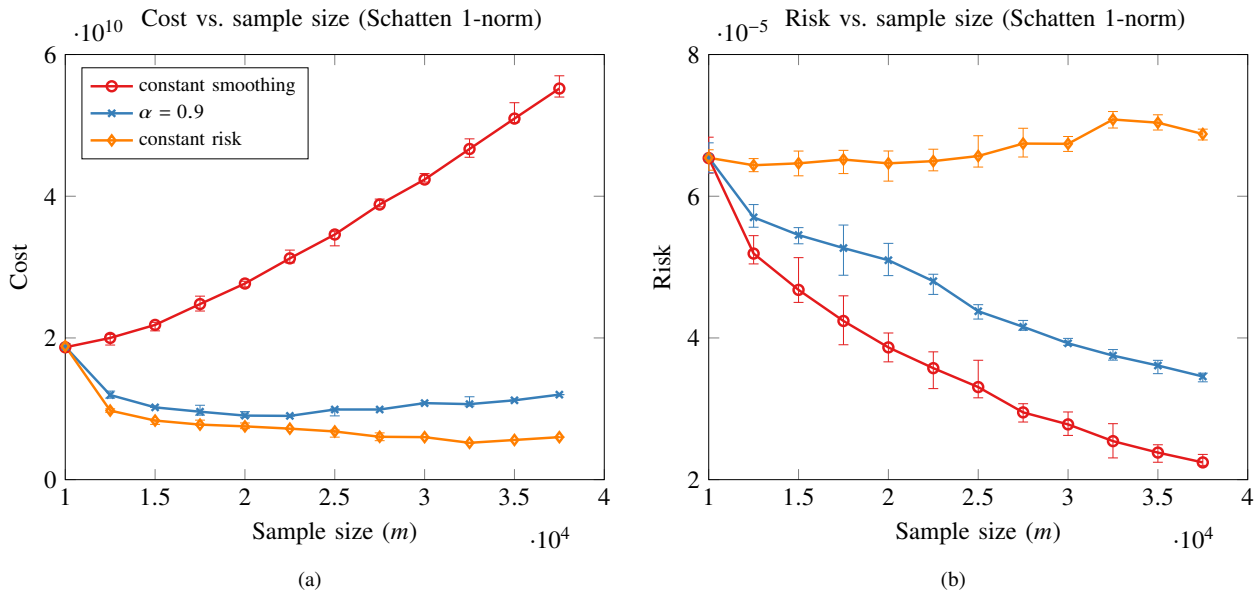
C. Numerical Experiment

Fig. 3 shows the results of a substantially similar numerical experiment to the one performed for sparse vector regression. Again, current practice dictates using a smoothing parameter that has no dependence on the sample size m; see [31], for example. In our tests, we choose the baseline smoothing parameter $\overline{\mu} = 0.1$ recommended by [27]. As before, we compare the constant smoothing, constant risk, and balanced ($\alpha = 0.9$) schemes. See Appendix A-B for the methodological details.

In this case, we use the ambient dimension $d = 200 \times 200$ and set the normalized rank $\rho = 5\%$. We test each method with 10 random trials of the low-rank matrix regression problem for each value of the sample size $m = 10\,000, 12\,500, 15\,000, \ldots, 37\,500$. The baseline sample size $\overline{m} = 10\,000$ corresponds roughly to $\delta(\mathcal{D}(f_{\overline{\mu}}; \boldsymbol{X}^\natural)) + \sqrt{200 \cdot 200}$, where $\boldsymbol{X}^\natural$ is the true random low-rank matrix.

The random measurement matrices are again partial unitary, and the nonzero singular values of the random low-rank matrices $\boldsymbol{X}^\natural$ are 1. We solve each problem with Algorithm 1 using the same relative primal feasibility gap tolerance of 10^{-3} as the stopping criterion. In this case, the statistical dimension bound given in Proposition VII.1 overestimates the risk incurred by a small amount. Therefore, we have not included the theoretical risk levels in Fig. 3(b), but the calculations still have value in determining an appropriate value of the scaling parameter α and, thereby, in computing the smoothing parameter μ.

In Fig. 3, we see the same qualitative behavior as in the sparse vector case. The constant smoothing scheme decreases risk the most over the range of sample sizes, but its cost continues to rise as the number of samples increases. The constant risk scheme provides the largest computational speedup but provides no improvement in statistical accuracy. The balanced method, however, achieves a 1.6× reduction in total computational cost from $m = 10\,000$ to $m = 37\,500$ while reducing risk by a factor of 1.9×. The observed speedup over the constant smoothing



(a) (b)

Fig. 3. **Low-rank matrix regression experiment.** The panels show (a) the average computational cost and (b) the estimated statistical risk over 10 random trials of the dual-smoothed low-rank matrix regression problem with ambient dimension $d = 200 \times 200$, normalized rank $\rho = 5\%$, and noise level $\sigma = 0.01$ for various sample sizes m. The red curve (circles) represents using a fixed smoothing parameter $\mu = 0.1$, the orange curve (diamonds) results from adjusting the smoothing parameter μ to maintain the baseline risk, and the blue curve (crosses) uses the balanced scheme (12) with scaling parameter $\alpha = 0.9$. For all schemes, the baseline smoothing parameter $\overline{\mu} = 0.1$, and the baseline sample size $\overline{m} = 10\,000$. The error bars indicate the minimum and maximum observed values.

scheme at $m = 37\,500$ is 4.6× while incurring statistical risk only 1.5× greater.

VIII. Example: Image Interpolation

In this section, we apply our tradeoff principle to an image interpolation problem.

A. The Optimization Problem

We let $\boldsymbol{X}^\natural$ be the matrix of pixel intensities of a grayscale image, and we observe the vector of pixels $\boldsymbol{b} := \mathcal{A}(\boldsymbol{X}^\natural)$, where $\mathcal{A}$ is the linear operator that returns a vector of m specific pixels from the original image. We assume that we know the operator $\mathcal{A}$, and therefore, we know the locations of the pixels being subsampled. To reconstruct the full image from the subsampled image, we solve

$$\begin{aligned} \text{minimize} \quad & \|\mathcal{W}(\boldsymbol{X})\|_{\ell_1} + \frac{\mu}{2}\|\mathcal{W}(\boldsymbol{X})\|^2 \\ \text{subject to} \quad & \mathcal{A}(\boldsymbol{X}) = \boldsymbol{b}, \end{aligned}$$

where $\mathcal{W}$ is the two-dimensional discrete cosine transformation with vectorized output. We use TFOCS [32], [22] to solve this dual-smoothed ℓ_1-analysis problem, and we use the Spot Toolbox[2] to implement the linear operators $\mathcal{W}$ and $\mathcal{A}$.

B. Numerical Experiment

Fig. 4 shows the results of an image interpolation experiment on a grayscale image of size 2867×1906 pixels. We solved the interpolation problem for pairs (ρ, μ) of the sampling density and smoothing parameter. At each iteration, we record the peak signal-to-noise ratio (PSNR) of the current iterate $\boldsymbol{X}_k$, where

$$\text{PSNR}(\boldsymbol{X}_k) = 10 \cdot \log_{10}\left(\frac{d_1 d_2}{\|\boldsymbol{X}_k - \boldsymbol{X}^\natural\|_{\text{F}}}\right), \tag{15}$$

and $\boldsymbol{X}^\natural$ is the original image of size $d_1 \times d_2$ pixels. See Appendix A-C for the methodological details.

The frontiers shown in solid blue lines indicate the Pareto efficient allocations of sample size and computational time for three different levels of PSNR. These allocations are achieved by employing additional smoothing as the sampling density increases. That is, as ρ increases, we can increase the smoothing parameter μ to achieve the same level of accuracy in the reconstructed image faster. The dashed red line, on the other hand, shows the result if we keep a constant smoothing parameter of $\mu = 0.02$ throughout the run of the experiment.

Without aggressive smoothing, the computational cost to reach the desired level of accuracy is several times higher. Indeed, for a reconstruction quality of 32 dB PSNR and a sampling density of $\rho = 40\%$, using the smoothing parameter $\mu = 0.02$ is 3.6× slower than using $\mu = 0.32$. This illustrates the benefit of smoothing these optimization problems more and more aggressively as the sampling density increases.

IX. Conclusion

The examples we have presented indeed show time–data tradeoffs achieved through dual-smoothing. We believe that our method can be used to show tradeoffs in other statistical problems solved using convex optimization. The key is understanding precisely how manipulating the geometry of the optimization problem affects the accuracy of the solution. This allows for the determination of whether the speedup

[2]http://www.cs.ubc.ca/labs/scl/spot/

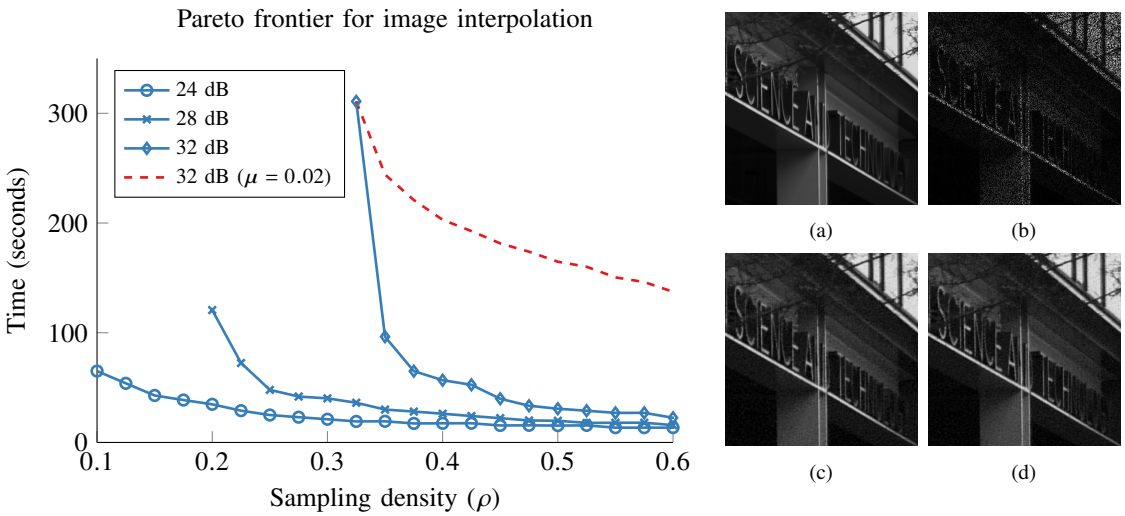


Fig. 4. **Image interpolation.** The graph shows the observed Pareto frontier in our image interpolation experiment where we treat the sampling density ρ and computational time as the two resources that we trade off. The solid blue lines give the Pareto frontiers achieved by aggressively smoothing the problem as we increase the sampling density ρ. These frontiers correspond to three different accuracy levels of the reconstructed images given as a peak signal-to-noise-ratio (PSNR). The dashed red line shows the frontier achieved for 32 dB PSNR accuracy with a fixed smoothing parameter $\mu = 0.02$ as sampling density ρ increases. Our aggressive smoothing outperforms the constant smoothing by a large margin. The grid of images shows 450×450 pixel patches of: (a) the original image, (b) the original image subsampled at $\rho = 40\%$, (c) the reconstructed image with $\rho = 40\%$ and $\mu = 0.02$ (32.1 dB PSNR), and (d) the reconstructed image with $\rho = 40\%$ and $\mu = 0.32$ (32.2 dB PSNR). The shown reconstructions are of the same quality despite the differing values of μ.

from a computationally beneficial relaxation is worth the loss of statistical accuracy. At the moment, this process may require some amount of numerical experimentation. As the geometric understanding of these optimization problems increases, however, we envision a richer theory of similar tradeoffs.

Appendix A
Numerical Methodology

This section describes the numerical experiments presented in Sections VI-C, VII-C, and VIII-B. All of the experiments discussed herein were performed on a 12-core workstation under MATLAB 2014a and OS X 10.9.5.

A. Sparse Vector Regression

The data for the sparse vector regression experiment in Section VI-C were generated as follows. Fix the ambient dimension $d = 40\,000$. For each of the smoothing schemes described in Section V-C (using $\alpha = 0.9$ in the balanced scheme) and each value of the sample size $m = 10\,000, 12\,000, 14\,000, \ldots, 38\,000$, perform 10 trials of this procedure, and average the results:

- Generate a sparse vector $\boldsymbol{x}^\natural$ with 2000 nonzero entries placed uniformly at random, each taking the value either -1 or $+1$ independently with equal probability.
- Choose a measurement matrix $\boldsymbol{A} \in \mathbb{R}^{m \times 40\,000}$ uniformly at random from the ensemble of $m \times 40\,000$ matrices with orthonormal rows (see [33] for the numerical details, as some care must be taken to ensure the appropriate distribution).
- Set the baseline smoothing parameter $\overline{\mu} = 0.1$ and the baseline sample size $\overline{m} = 10\,000$, which is approximately $\delta(\mathcal{D}(f_{\overline{\mu}}; \boldsymbol{x}^\natural)) + \sqrt{40\,000}$.
- Calculate the smoothing parameter μ according to the current scheme and the resulting statistical dimension $\delta = \delta(\mathcal{D}(f_\mu; \boldsymbol{x}^\natural))$.
- Set the parameter $\epsilon = \sigma(m - \delta)^{1/2}$.
- Use the Auslender–Teboulle algorithm (Algorithm 1) to solve the dual-smoothed sparse vector regression problem.
- Stop the algorithm when the relative primal feasibility gap $|\|\boldsymbol{A}\boldsymbol{x}_k - \boldsymbol{b}\| - \epsilon| / \epsilon < 10^{-3}$.
- Store the computational cost $k \cdot m \cdot 40\,000$ and the (average) squared prediction error $\|\boldsymbol{A}(\widehat{\boldsymbol{x}} - \boldsymbol{x}^\natural)\|^2 / m$, where $\widehat{\boldsymbol{x}}$ is the final value of the primal iterate.

B. Low-rank Matrix Regression

The data for the low-rank matrix regression experiment in Section VII-C were generated as follows. Fix the ambient dimensions $d = d_1 d_2 = 200 \cdot 200 = 40\,000$. For each of the smoothing schemes described in Section V-C (using $\alpha = 0.9$ in the balanced scheme) and each value of the sample size $m = 10\,000, 12\,500, 15\,000, \ldots, 37\,500$, perform 10 trials of this procedure, and average the results:

- Generate a low-rank matrix $\boldsymbol{X}^\natural := \boldsymbol{Q}_1 \boldsymbol{Q}_2^T$, where the $\boldsymbol{Q}_i$ are chosen uniformly at random from the ensemble of 200×10 matrices with orthonormal columns.
- Choose a measurement matrix $\boldsymbol{A} \in \mathbb{R}^{m \times 40\,000}$ uniformly at random from the ensemble of $m \times 40\,000$ matrices with orthonormal rows.
- Set the baseline smoothing parameter $\overline{\mu} = 0.1$ and the baseline sample size $\overline{m} = 10\,000$, which is approximately $\delta(\mathcal{D}(f_{\overline{\mu}}; \boldsymbol{x}^\natural)) + \sqrt{40\,000}$.
- Calculate the smoothing parameter μ according to the current scheme and the resulting statistical dimension $\delta = \delta(\mathcal{D}(f_\mu; \boldsymbol{x}^\natural))$.
- Set the parameter $\epsilon = \sigma(m - \delta)^{1/2}$.

- Use the Auslender–Teboulle algorithm (Algorithm 1) to solve the dual-smoothed low-rank matrix regression problem.
- Stop the algorithm when the relative primal feasibility gap $|\|\boldsymbol{A}\cdot\mathrm{vec}(\boldsymbol{X}_k)-\boldsymbol{b}\|-\epsilon|/\epsilon<10^{-3}$.
- Store the computational cost $k\cdot m\cdot 40\,000$ and the (average) squared prediction error $\|\boldsymbol{A}\cdot\mathrm{vec}(\widehat{\boldsymbol{X}}-\boldsymbol{X}^{\natural})\|^2/m$, where $\widehat{\boldsymbol{X}}$ is the final value of the primal iterate.

C. Image Interpolation

The data for the image interpolation experiment in Section VIII-B were generated as follows. We loaded a 16-bit grayscale TIFF image of size $d = 2867\times 1906$ pixels into MATLAB. For each sampling density $\rho = 10\%, 12.5\%, \ldots, 97.5\%$, we performed the following procedure for each of the smoothing parameters $\mu = 0.01, 0.02, 0.04, \ldots, 5.12$:

- Choose an ordered subset of ρd pixels from the image uniformly at random.
- Using the Spot Toolbox, construct both the subsampling operator $\mathcal{A}$ that returns the random (ordered) subset of pixels as a column vector and the 2D DCT operator $\mathcal{W}$ (`opDCT2`).
- Generate the subsampled observations $\boldsymbol{b}$ by applying $\mathcal{A}$ to the image.
- Use the TFOCS solver `solver_sBPDN_W` to solve the ℓ_1-analysis problem with parameters $\epsilon = 0$ and μ. Set the TFOCS options to tell the solver that $\|\mathcal{A}\|^2 = \|\mathcal{W}\|^2 = 1$. Use the history feature of TFOCS to record the peak signal-to-noise ratio (15) of each iterate, and set the solver to stop after 200 iterations or when the observed PSNR is greater than 40 dB.
- Record the time taken, the history of PSNRs, and the number of iterations completed.

Appendix B
Proof of Theorem IV.5

This appendix provides the proof used to bound the feasibility gap of the primal iterates as a function of the number of iterations taken.

Proof of Theorem IV.5: Let g be the dual function (9). Define $G := -g$, $\tilde{G} = -\tilde{g}$, and $H = -h$, so that $G, \tilde{G}, H$ are convex. By Lemma IV.4, the function $\tilde{G}$ has a Lipschitz continuous gradient with parameter L_μ. Therefore, it has a quadratic upper bound, and we find

$$\begin{aligned} G(z^\star) = \inf_z G(z) &= \inf_z(\tilde{G}(z)+H(z)) \\ &\le \inf_z\left\{\tilde{G}(\boldsymbol{y}_k)+\left\langle\nabla\tilde{G}(\boldsymbol{y}_k),\ z-\boldsymbol{y}_k\right\rangle\right. \\ &\qquad \left.+\frac{L_\mu}{2}\|z-\boldsymbol{y}_k\|^2+H(z)\right\}, \end{aligned}$$

for any $k \ge 0$. Note that this quantity is a composite gradient mapping and, for our choice of H, equals

$$\begin{aligned} J(\bar{z}_{k+1}) := \tilde{G}(\boldsymbol{y}_k)+\left\langle\nabla\tilde{G}(\boldsymbol{y}_k),\ \bar{z}_{k+1}-\boldsymbol{y}_k\right\rangle \\ +\frac{L_\mu}{2}\|\bar{z}_{k+1}-\boldsymbol{y}_k\|^2+H(z_{k+1}), \end{aligned}$$

where

$$\bar{z}_{k+1} = \mathrm{Shrink}\left(\boldsymbol{y}_k - L_\mu^{-1}\nabla\tilde{G}(\boldsymbol{y}_k), \frac{\epsilon}{L_\mu}\right).$$

Now since J is L_μ-strongly convex in $\bar{z}_{k+1}$, it has the quadratic lower bound

$$J(\boldsymbol{y}_k) - J(\bar{z}_{k+1}) = G(\boldsymbol{y}_k) - J(\bar{z}_{k+1}) \ge \frac{L_\mu}{2}\|\boldsymbol{y}_k-\bar{z}_{k+1}\|^2.$$

We then have

$$G(z^\star) \le J(\bar{z}_{k+1}) \le G(\boldsymbol{y}_k) - \frac{L_\mu}{2}\|\boldsymbol{y}_k-\bar{z}_{k+1}\|^2,$$

and so

$$\frac{L_\mu}{2}\|\boldsymbol{y}_k-\bar{z}_{k+1}\|^2 \le G(\boldsymbol{y}_k) - G(z^\star) = g(z^\star) - g(\boldsymbol{y}_k).$$

By the definition of the Shrink operator, the dual iterate $\bar{z}_{k+1}$ may take on either of two values. When

$$\|\boldsymbol{y}_k - L_\mu^{-1}\nabla\tilde{G}(\boldsymbol{y}_k)\| \le \frac{\epsilon}{L_\mu},$$

$\bar{z}_{k+1} = \boldsymbol{0}$. An application of the reverse triangle inequality and some rearranging gives that

$$\frac{1}{L_\mu}(\|\nabla\tilde{G}(\boldsymbol{y}_k)\|-\epsilon) \le \|\boldsymbol{y}_k\| = \|\boldsymbol{y}_k-\bar{z}_{k+1}\|,$$

and so

$$\left|\,\|\nabla\tilde{G}(\boldsymbol{y}_k)\|-\epsilon\right|^2 \le L_\mu^2\cdot\|\boldsymbol{y}_k-\bar{z}_{k+1}\|^2.$$

Otherwise, the iterate $\bar{z}_{k+1}$ takes the value

$$\left(1-\frac{\epsilon}{L_\mu\|\boldsymbol{y}_k-L_\mu^{-1}\nabla\tilde{G}(\boldsymbol{y}_k)\|}\right)\cdot(\boldsymbol{y}_k-L_\mu^{-1}\nabla\tilde{G}(\boldsymbol{y}_k)),$$

and we can compute

$$L_\mu^2\cdot\|\boldsymbol{y}_k-\bar{z}_{k+1}\|^2 = \left\|\nabla\tilde{G}(\boldsymbol{y}_k)+\epsilon\left(\frac{\boldsymbol{y}_k-L_\mu^{-1}\nabla\tilde{G}(\boldsymbol{y}_k)}{\|\boldsymbol{y}_k-L_\mu^{-1}\nabla\tilde{G}(\boldsymbol{y}_k)\|}\right)\right\|^2.$$

By the reverse triangle inequality, we have

$$\left|\,\|\nabla\tilde{G}(\boldsymbol{y}_k)\|-\epsilon\right|^2 \le L_\mu^2\cdot\|\boldsymbol{y}_k-\bar{z}_{k+1}\|^2,$$

the exact same bound as above. Therefore, we can conclude that for all $k \ge 0$,

$$\left|\,\|\boldsymbol{A}\boldsymbol{x}_k-\boldsymbol{b}\|-\epsilon\right|^2 \le \frac{2\|\boldsymbol{A}\|^2}{\mu}\cdot\left(g(z^\star)-g(\boldsymbol{y}_k)\right),$$

where we use Lemma IV.4 to substitute for L_μ and $\nabla\tilde{G}(\boldsymbol{y}_k)$.

The quantity on the right is bounded by the standard convergence result for the Auslender–Teboulle algorithm (see [21, Thm 5.2] and [15, Coro. 1]):

$$g(z^\star) - g(\boldsymbol{y}_k) \le \frac{2L_\mu\|z^\star\|^2}{k^2} = \frac{2\|\boldsymbol{A}\|^2\|z^\star\|^2}{\mu\cdot k^2}.$$

We rearrange terms and take the square root to complete the proof. ∎

REFERENCES

[1] S. Shalev-Shwartz and N. Srebro, "SVM optimization: inverse dependence on training set size," in *Proc. 25th Annu. Int. Conf. Machine Learning (ICML 2008)*, pp. 928–935, ACM, 2008.

[2] J. J. Bruer, J. A. Tropp, V. Cevher, and S. R. Becker, "Time–Data Tradeoffs by Aggressive Smoothing," in *Advances in Neural Information Processing Systems 27 (NIPS 2014)*, pp. 1664–1672, 2014.

[3] D. Amelunxen, M. Lotz, M. B. McCoy, and J. A. Tropp, "Living on the edge: A geometric theory of phase transitions in convex optimization," *Information and Inference*, vol. to appear, 2014.

[4] S. Oymak and B. Hassibi, "Sharp MSE Bounds for Proximal Denoising," *arXiv*, 2013, 1305.2714v5.

[5] L. Bottou and O. Bousquet, "The tradeoffs of large scale learning," in *Advances in Neural Information Processing Systems 20 (NIPS 2007)*, pp. 161–168, 2008.

[6] A. Agarwal, P. L. Bartlett, and J. C. Duchi, "Oracle inequalities for computationally adaptive model selection," *arXiv*, 2012, 1208.0129v1.

[7] S. Shalev-Shwartz, O. Shamir, and E. Tromer, "Using More Data to Speed-up Training Time," in *Proc. 15th Int. Conf. Artificial Intelligence and Statistics*, pp. 1019–1027, 2012.

[8] Q. Berthet and P. Rigollet, "Computational Lower Bounds for Sparse PCA," *arXiv*, 2013, 1304.0828v2.

[9] A. Daniely, N. Linial, and S. Shalev-Shwartz, "More data speeds up training time in learning halfspaces over sparse vectors," in *Advances in Neural Information Processing Systems 26 (NIPS 2013)*, pp. 145–153, 2013.

[10] D. Shender and J. Lafferty, "Computation-Risk Tradeoffs for Covariance-Thresholded Regression," in *Proc. 30th Int. Conf. Machine Learning (ICML 2013)*, pp. 756–764, 2013.

[11] M. I. Jordan, "On statistics, computation and scalability," *Bernoulli*, vol. 19, no. 4, pp. 1378–1390, 2013.

[12] V. Chandrasekaran and M. I. Jordan, "Computational and statistical tradeoffs via convex relaxation," *Proc. Natl. Acad. Sci. USA*, vol. 110, no. 13, pp. E1181–E1190, 2013.

[13] A. S. Nemirovsky and D. B. Yudin, *Problem complexity and method efficiency in optimization*. A Wiley-Interscience Publication, New York: John Wiley & Sons Inc., 1983.

[14] Y. E. Nesterov, "A method for solving the convex programming problem with convergence rate O($1/k^2$)," *Dokl. Akad. Nauk SSSR*, vol. 269, no. 3, pp. 543–547, 1983.

[15] P. Tseng, "On Accelerated Proximal Gradient Methods for Convex-Concave Optimization," tech. rep., Department of Mathematics, University of Washington, Seattle, WA, May 2008.

[16] Y. Nesterov, "Smooth minimization of non-smooth functions," *Math. Program.*, vol. 103, no. 1, pp. 127–152, 2005.

[17] A. Beck and M. Teboulle, "Smoothing and first order methods: a unified framework," *SIAM J. Optim.*, vol. 22, no. 2, pp. 557–580, 2012.

[18] R. T. Rockafellar and R. J.-B. Wets, *Variational Analysis*. New York: Springer, 1997.

[19] Y. Nesterov, "Gradient Methods for Minimizing Composite Objective Function," Tech. Rep. 2007/76, CORE, Université catholique de Louvain, Louvain-la-Neuve, Belgium, 2007.

[20] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge: Cambridge University Press, 2004.

[21] A. Auslender and M. Teboulle, "Interior gradient and proximal methods for convex and conic optimization," *SIAM J. Optim.*, vol. 16, no. 3, pp. 697–725, 2006.

[22] S. R. Becker, E. J. Candès, and M. C. Grant, "Templates for convex cone problems with applications to sparse signal recovery," *Math. Program. Comput.*, vol. 3, no. 3, pp. 165–218, 2011.

[23] Q. Tran-Dinh and V. Cevher, "Constrained convex minimization via model-based excessive gap," in *Advances in Neural Information Processing Systems 27 (NIPS 2014)*, pp. 721–729, 2014.

[24] M. B. McCoy and J. A. Tropp, "The achievable performance of convex demixing," *arXiv*, Sept. 2013, 1309.7478v1.

[25] R. Tibshirani, "Regression shrinkage and selection via the Lasso," *J. R. Stat. Soc. Ser. B Stat. Methodol.*, vol. 58, no. 1, pp. 267–288, 1996.

[26] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *J. R. Stat. Soc. Ser. B Stat. Methodol.*, vol. 67, pp. 301–320, 2005.

[27] M.-J. Lai and W. Yin, "Augmented l(1) and Nuclear-Norm Models with a Globally Linearly Convergent Algorithm," *SIAM J. Imaging Sci.*, vol. 6, no. 2, pp. 1059–1091, 2013.

[28] J.-F. Cai, S. Osher, and Z. Shen, "Linearized Bregman Iterations for Compressed Sensing," *Math. Comp.*, vol. 78, no. 267, pp. 1515–1536, 2009.

[29] S. Osher, Y. Mao, B. Dong, and W. Yin, "Fast linearized Bregman iteration for compressive sensing and sparse denoising," *Commun. Math. Sci.*, vol. 8, no. 1, pp. 93–111, 2010.

[30] K.-C. Toh and S. Yun, "An accelerated proximal gradient algorithm for nuclear norm regularized linear least squares problems," *Pac. J. Optim.*, vol. 6, no. 3, pp. 615–640, 2010.

[31] J.-F. Cai, E. J. Candès, and Z. Shen, "A singular value thresholding algorithm for matrix completion," *SIAM J. Optim.*, vol. 20, no. 4, pp. 1956–1982, 2010.

[32] S. Becker, E. J. Candès, and M. Grant, "TFOCS v1.2 user guide," 2012.

[33] F. Mezzadri, "How to generate random matrices from the classical compact groups," *Notices Amer. Math. Soc.*, vol. 54, no. 5, pp. 592–604, 2007.

John J. Bruer received the B.A. (*summa cum laude*) degree in Mathematics from New York University, New York, NY, USA, in 2008. He is currently a candidate for the Ph.D. degree in Applied and Computational Mathematics at the California Institute of Technology, Pasadena, CA, USA. His research interests include optimization, machine learning, and statistics.

Joel A. Tropp is Professor of Applied & Computational Mathematics at the California Institute of Technology. He earned the Ph.D. degree in Computational Applied Mathematics from the University of Texas at Austin in 2004. Prof. Tropp's interests lie at the interface of applied mathematics, electrical engineering, computer science, and statistics. His work has been recognized with the 2008 PECASE, the EUSIPCO 2010 Best Paper Award, and the 2011 SIAM Outstanding Paper Prize. He is a 2014 Thomson Reuters Highly Cited Researcher in Computer Science.

Volkan Cevher received the B.S. (valedictorian) degree in electrical engineering in 1999 from Bilkent University in Ankara, Turkey, and he received the Ph.D. degree in Electrical and Computer Engineering in 2005 from the Georgia Institute of Technology in Atlanta. He held research scientist positions at the University of Maryland, College Park from 2006 to 2007 and at Rice University in Houston, Texas, from 2008 to 2009. Currently, he is an Assistant Professor at the Swiss Federal Institute of Technology Lausanne with a complimentary Faculty Fellow appointment at the Electrical and Computer Engineering Department at Rice University. His research interests include signal processing, optimization, machine learning, and information theory. He received a Best Paper Award at SPARS in 2009 and an ERC StG in 2011.

Stephen R. Becker is an assistant professor in the Applied Math department at the University of Colorado at Boulder. Previously he was a Goldstine Postdoctoral fellow at IBM Research T. J. Watson lab and a postdoctoral fellow at the Laboratoire Jacques-Louis Lions at Paris 6 University. He received his Ph.D. in Applied and Computational Mathematics from the California Institute of Technology in 2011, and bachelor degrees in Math and Physics from Wesleyan University in 2005. His work focuses on large-scale continuous optimization for signal processing and machine learning applications.