Safeguarded Learned Convex Optimization

Howard Heaton*[1] Xiaohan Chen*[2] Atlas Wang[3] Wotao Yin[2]

[1]Typal Research, Typal LLC [2]Alibaba US, DAMO Academy, Decision Intelligence Lab [3]Department of Electrical and Computer and Engineering, The University of Texas at Austin

Overview

Applications abound in which optimization problems must be repeatedly solved, each time with new (but similar) data. Analytic optimization algorithms can be hand-designed to provably solve these problems in an iterative fashion. On one hand, data-driven algorithms can "learn to optimize" (L2O) with much fewer iterations and similar cost per iteration as general purpose optimization algorithms. On the other hand, many L2O algorithms lack converge guarantees.

To fuse the advantages of these approaches, we present a Safe-L2O framework. Safe-L2O updates incorporate a safeguard to guarantee convergence for convex problems with proximal and/or gradient oracles. The safeguard is simple and computationally cheap to implement, and it is activated only when the data-driven L2O updates would perform poorly or appear to diverge. This yields the numerical benefits of employing machine learning to create rapid L2O algorithms while still guaranteeing convergence. Our numerical examples show convergence of Safe-L2O algorithms, even when the provided data is not from the distribution of training data.

Key Problem

Given data d, consider the optimization problem

$$\min_{x\in\mathbb{R}^n} f(x;\ d). \tag{1}$$

We aim to solve this problem rapidly and repeatedly, each time with new, but similar data. Machine learning tools can enable rapid computation of approximate solutions, but most lack strong guarantees.

Can a safeguard be added to L2O algorithms to give convergence without significantly hindering performance?

This work, Safe-L2O, answers the above affirmatively.

Learning to Optimize (L2O)

Neural networks can be designed to generalize optimization algorithms. These are called L2O models, which exhibit great success numerically, in some cases speeding up convergence by *multiple orders of magnitude* (*e.g.* [2]).

Check out the L2O survey [1] for more details.

Check out our video on YouTube: "Safeguarded Learned Convex Optimization"

Safeguards

Iterative optimization algorithms with L2O can be safeguarded via

$$x^{k+1} = \begin{cases} \text{L2O Update} & \text{if L2O Update is "good"} \\ \text{Classic Update otherwise.} \end{cases} \tag{2}$$

We seek to design a "good" safeguard with the following properties.

1. The safeguard should ensure certain forms of worst-case convergence similar to analytic algorithms.
2. The safeguard must only use known quantities related to convex problems (*e.g.* objective values, gradient norms)
3. Both L2O and Safe-L2O schemes should perform identically on "good" data with comparable per-iteration costs.
4. The safeguard should kick in only when "bad" L2O updates would otherwise occur.

Many optimization algorithms (*e.g.* proximal gradient, ADMM, Douglas Rachford splitting) may be written via an update operator T and a sequence $\{x^k\}$ defined by

$$x^{k+1} = T(x^k;\ d), \quad \text{for all } k \in \mathbb{N}. \tag{3}$$

For example, with gradient descent, $T(x) = x - \lambda\nabla f(x)$ for some step size $\lambda > 0$. As $\{x^k\}$ converges to a solution, the fixed point residual converges to zero, *i.e.* $\|x^k - T(x^k; d)\| \to 0$.

Our safeguard uses the fixed point residual. For $\beta > 0$, we define

$$\text{Residual}(y;\ x^k) = \|y - T(y;\ d)\| + \beta\|y - x^k\|. \tag{4}$$

For $\alpha \in (0,1)$, we say

$$(y \text{ is good}) = \text{Residual}(y;\ x^k) \le \alpha\mu_k. \tag{5}$$

We let y^{k+1} be the L2O update for x^k; if it is good, then $x^{k+1} = y^{k+1}$. The safeguard sequence $\{\mu_k\}$ updates with $\theta \in (0,1)$ by

$$\mu_{k+1} = \begin{cases} (1-\theta)\cdot\mu_k + \theta\cdot\text{Residual}(y^{k+1};\ x^k) & \text{if } y^{k+1} \text{ is good} \\ \mu_k & \text{otherwise.} \end{cases} \tag{6}$$

Safe L2O Guarantees

(Informal) Convergence Result. If a sequence $\{x^k\}$ is generated by our Safe-L2O scheme (2) with one of our safeguard schemes and a classic operator T, then it converges to converges to a solution to (1), *i.e.*

$$\lim_{k\to\infty} x^k = x_d^\star \in \arg\min_{x\in\mathbb{R}^n} f(x;\ d). \tag{7}$$

Experiments

Here for a short and fat dictionary A, we minimize

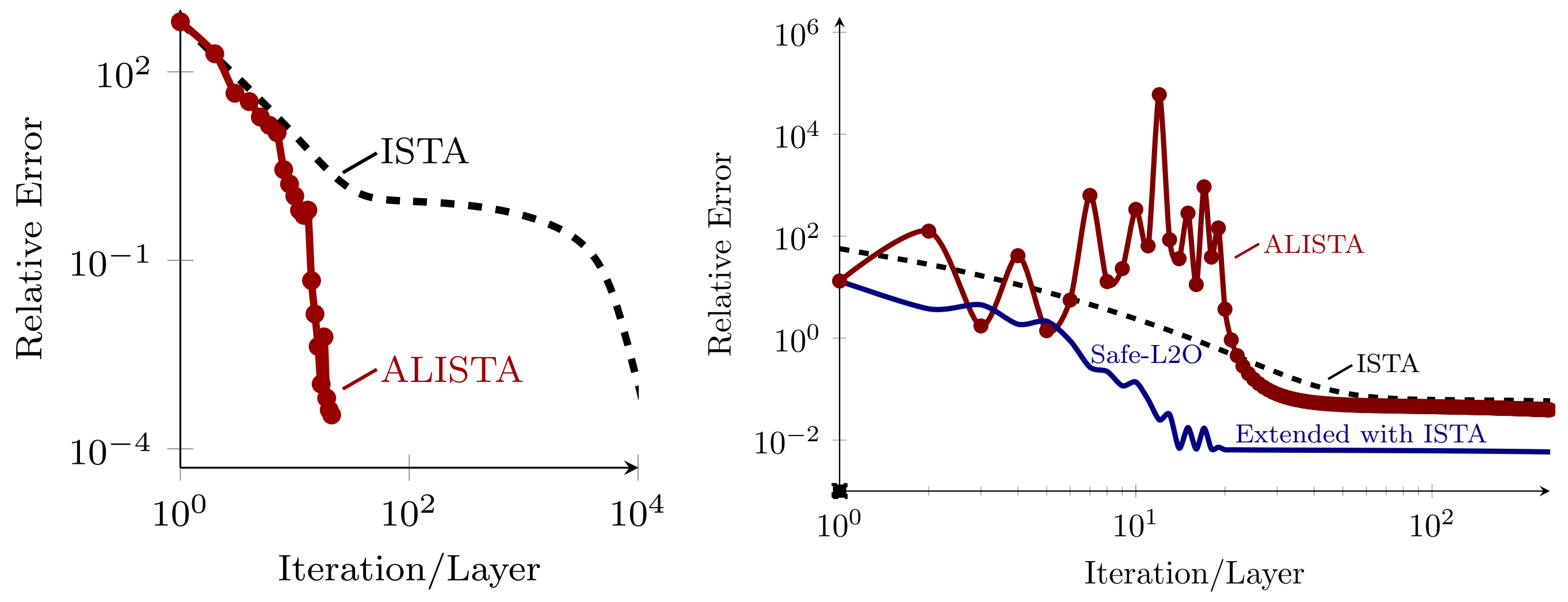
$$\frac{1}{2}\|Ax - d\|_2^2 + \tau\|x\|_1, \tag{8}$$

where $x_d^\star$ is a sparse signal that gneerates data $d = Ax_d^\star + \varepsilon$, where ε is noise. For AdaLista, the dictionary A can also vary.

Goal: Show L2O schemes "break," but Safe-L2O always converges.

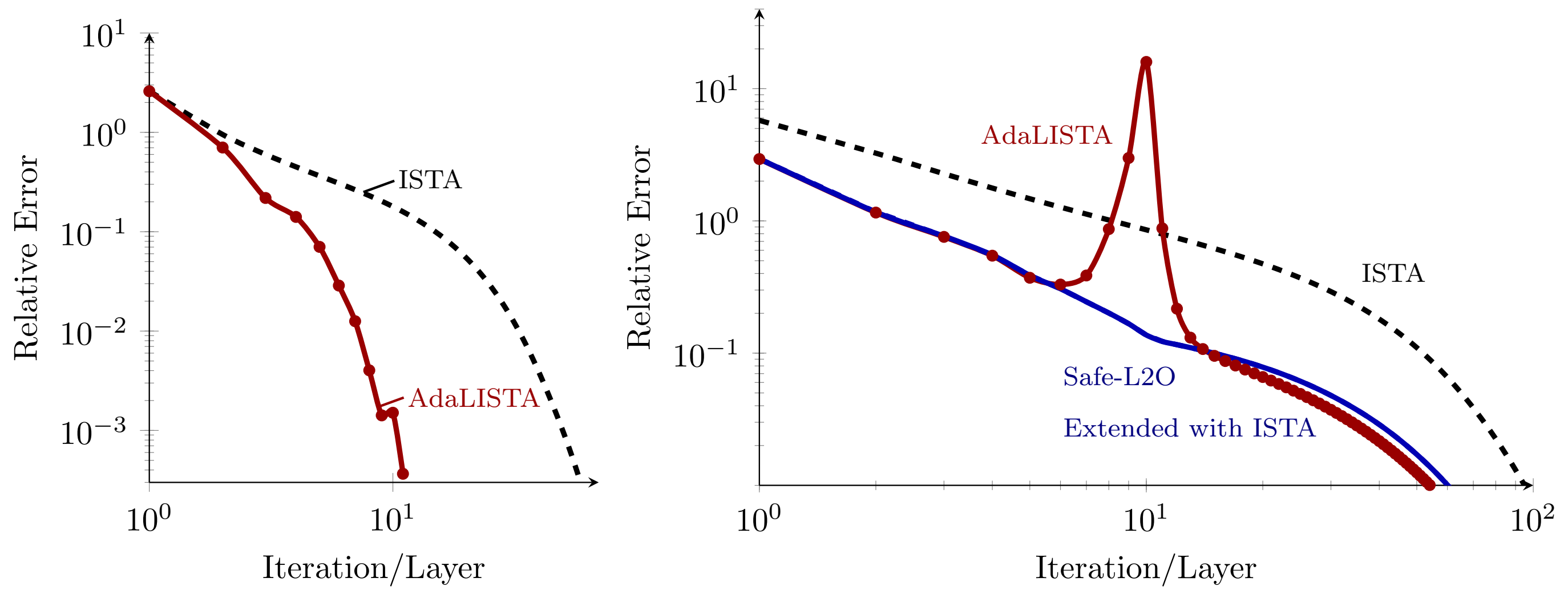
Seen Distribution: Distribution of training data*
(Often train/test data are same distribution, with different samples)

Unseen Distribution: Data generated from different distribution, (*e.g.* $x_d^\star$ that is *not* sparse)



(a) Seen Distribution (b) Unseen Distribution

Figure 1. ALISTA converges fast when d looks like training data, but diverges otherwise. Safeguarded ALISTA always converges.



(a) Seen Distribution (b) Unseen Distribution

Figure 2. AdaLISTA converges fast when d looks like training data, but diverges otherwise. Safeguarded AdaLISTA always converges.

References

[1] Tianlong Chen, Xiaohan Chen, Wuyang Chen, Howard Heaton, Jialin Liu, Zhangyang Wang, and Wotao Yin.
Learning to optimize: A primer and a benchmark.
Journal of Machine Learning Research, 2022.

[2] Jialin Liu and Xiaohan Chen.
Alista: Analytic weights are as good as learned weights in lista.
In *International Conference on Learning Representations (ICLR)*, 2019.

research.typal.llc/learning-to-optimize/safe-l2o AAAI Conference 2023, Washington D.C. xiaohan.chen@alibaba-inc.com