Online Linearized LASSO

Shuoguang Yang
IEDA, HKUST

Yuhao Yan
IEDA, HKUST

Xiuneng Zhu
Tower Research Capital LLC

Qiang Sun
University of Toronto

Abstract

Sparse regression has been a popular approach to perform variable selection and enhance the prediction accuracy and interpretability of the resulting statistical model. Existing approaches focus on offline regularized regression, while the online scenario has rarely been studied. In this paper, we propose a novel online sparse linear regression framework for analyzing streaming data when data points arrive sequentially. Our proposed method is memory efficient and requires less stringent restricted strong convexity assumptions. Theoretically, we show that with a properly chosen regularization parameter, the ℓ_2-error of our estimator decays to zero at the optimal order of $\widetilde{\mathcal{O}}(\sqrt{s/t})$, where s is the sparsity level, t is the streaming sample size, and $\widetilde{\mathcal{O}}(\cdot)$ hides logarithmic terms. Numerical experiments demonstrate the practical efficiency of our algorithm.

1 INTRODUCTION

With the development of modern data acquisition technologies, high-dimensional statistics has attracted significant interest due to its ability in handling datasets with large numbers of features. To alleviate the challenges introduced by massive amounts of features, a popular assumption is the sparsity assumption, that is, only a few variables contribute to the response. A common approach that exploits such sparsity assumption is to solve the following LASSO problem (Tibshirani, 1996)

$$\hat{\beta} = \arg\min_{\beta \in \mathbb{R}^p} \Big\{ \mathcal{L}(\beta) + \lambda \|\beta\|_1 \Big\}, \tag{1}$$

where $\mathcal{L}(\beta)$ is the empirical loss function, β is a p-dimensional coefficient vector, and $\lambda\|\beta\|_1$ is the LASSO penalty with λ being the regularization parameter.

Proceedings of the 26th International Conference on Artificial Intelligence and Statistics (AISTATS) 2023, Valencia, Spain. PMLR: Volume 206.

Studies in high-dimensional statistics can be divided into two major streams, the offline and online sparse problems, based on data sampling mechanisms. Originated from Tibshirani (1996), the offline problem considers an environment where data are available in its entirety at the beginning and the decision-maker aims to compute a good estimator by solving the corresponding regularized optimization problem. This problem have been previously studied from both statistical and computational perspectives (Agarwal et al., 2012; Loh and Wainwright, 2015; Fan et al., 2018b).

In contrast, online sparse regression considers the environment where data are not readily accessible at the beginning but arrive sequentially. Instead of solving for one final estimator in the offline setting, the online problem requires computing a sequence of estimators $\{\hat{\beta}_t\}$ such that the working estimator can be computed efficiently whenever additional samples arrive. This requires a significantly amount of effort and has been less studied in high dimensional statistics (Kale, 2014; Kale et al., 2017; Fan et al., 2018a).

In this paper, we consider the online environment where a feature-label pair (x_t, y_t) comes in each round $t \geq 1$. Here $x_t \in \mathbb{R}^p$ is the covariate and y_t is the response depending on both x_t and an unknown coefficient $\beta^* \in \mathbb{R}^p$. Letting $S = \text{support}(\beta^*)$ be the support of β^*, we assume β^* preserves a sparse structure whose sparsity level, or the cardinality of support, s is much smaller than the dimension p, i.e., $s = |S| = \|\beta^*\|_0 \ll p$. Our goal is to compute an estimator β_t in each round t to estimate the underlying sparse coefficient β^*.

Although an extensive amount of effort has been made to study online optimization in the low-dimensional setting (Kushner and Yin, 1997; Rakhlin et al., 2011), online high-dimensional sparse optimization has been much less understood, due to the extra sparse-inducing regularizer $\lambda\|\beta\|_1$. It mainly suffers from three key challenges. (i) Memory and storage challenge: Naively utilizing the entire dataset up to time t incurs a cost of $\mathcal{O}(pt)$ storage and memory complexity when computing the solution sequence. (ii) Dynamic update of the regularization parameter: To ensure the best statistical performance of the estimator sequence at each time t, the data-dependent regularization parameter

λ needs to be updated in each online round as the sample size t increases when new data arrive. Most sparse online optimization algorithms (Langford et al., 2009; Xiao, 2009) for regularized problems do not apply to our settings since they only consider the fixed-λ optimization problems. Part of the reason is that they did not consider optimal statistical guarantees such as the parameter estimation error or other optimality measures. (iii) Restricted strong convexity: In the online setting, to ensure each of the estimator sequence to be sparse, one needs the restricted strong convexity (RSC) condition to hold uniformly for all online learning rounds. This can be rather stringent in practice because the loss function varies in each round. Therefore, it remains a challenge whether a memory efficient framework could be developed for online sparse optimization with optimal statistical guarantees (Kale, 2014).

In this paper, we propose an online algorithm which in round t solves the following linearized LASSO problem

$$\hat{\beta}_t = \underset{\beta \in \mathbb{R}^p}{\arg\min} \Big\{ L_t(\beta; \hat{\beta}_{t-1}) + \lambda_t \|\beta\|_1 \Big\}.$$

where

$$\begin{aligned} & L_t(\beta; \hat{\beta}_{t-1}) \\ &= \underbrace{\ell_0(\beta)}_{\text{squared loss}} + \underbrace{\Big\langle \sum_{j=1}^{t} \frac{w_{t,j}}{W_t} \nabla \ell_j(\hat{\beta}_{t-1}) - \nabla \ell_0(\hat{\beta}_{t-1}), \beta \Big\rangle}_{\text{linearized loss}}, \end{aligned}$$

is the loss function, consisting of a squared loss $\ell_0(\beta)$ of an initial batch of sample size t_0, and a linearized loss evaluated at current best estimator $\hat{\beta}_{t-1}$ with data received before up to rounds t. Here, $\ell_j(\beta)$ is the squared loss incurred by the j-th data point, $w_{t,j}$ is the weight associated with $\nabla \ell_j(\hat{\beta}_{t-1})$, and $W_t = \sum_{j=1}^{t} w_{t,j}$. Notably, our framework enjoys a low memory cost of $\mathcal{O}(p\, t_0 + p^2)$ that does not depend on the online round t, and only requires the RSC condition to hold for the initial loss $\ell_0(\beta)$ instead of all online rounds. We show that under mild conditions, the entire solution trajectory $\{\hat{\beta}_t\}$ falls within a restricted cone, and is consistent such that it converges to the underlying coefficient vector at the best possible convergence rate. To our best knowledge, this is the first algorithm that enjoys the above properties.

Contributions. We make the following three major contributions.

(i) We propose a memory efficient scheme for online sparse linear regression that iteratively solves an LASSO optimization problem. Our framework adopts a novel loss function that consists of a squared loss of an initial batch of sample size t_0 and a linearized loss, which enjoys a fixed $\mathcal{O}(t_0 p + p^2)$ memory cost. Meanwhile, our resulting loss function satisfies the RSC condition throughout all online rounds as long as it holds for the initial batch. This is a significant boost as we do not need it to hold uniformly for all online learning rounds.

(ii) We show that by properly choosing the regularization parameters, our estimators are consistent whose ℓ_1-norm parameter estimation errors decay to zero in the optimal rate of $\tilde{\mathcal{O}}(\sqrt{s/t})$, where t is the streaming sample size and s is the sparsity level.

(iii) We conduct numerical experiments to test the practical performance of our algorithm against other baseline algorithms under various settings. Numerical results demonstrate the practical efficiency of our algorithm and validates our theoretical results.

1.1 Related Work

There has been relatively less work for online sparse regressions, which has different flavors from ours. We discuss those that are mostly related to our work.

Optimization perspective: As mentioned previously, previously sparse online optimization algorithms (Xiao, 2009; Langford et al., 2009; Bertsekas, 2011; Duchi et al., 2011; Yang et al., 2010) consider fixed-λ optimization problems and thus do not apply to our settings. Part of the reason is that they focused on regret bound for fixed λ and did not study the optimal statistical performance of the estimator sequence in a statistical setting. For the regularized optimization problem (2), the data-dependent regularized parameter has to be updated in each online round, and this makes the statistical analysis challenging. Garrigues and Ghaoui (2008) proposed a heuristic algorithm that conducts dynamic update of the regularization parameter but did not present any theoretical guarantees. Another approach that avoids this challenge is the online sparsity constrained optimization, which however brings additional computational intractabilities as it is commonly believed that sparsity contrained optimization is NP hard (Foster et al., 2015). Kale (2014) raised the open question that whether it is possible to design an efficient algorithm for the online sparse regression problem to achieve a sublinear regret bound. Toward addressing this challenge, Kale et al. (2017) proposed to solve a sequence of Dantzig selector problems in an online manner, which achieved a sublinear regret bound. However, their result requires a bound on the Restricted Isometry Property constant $\leq 1/5$ uniformly over all online rounds, or equivalently a bound on the condition number $\leq 3/2$ uniformly. This is undesirable because real-life high-dimensional data analyses routinely require estimation methods under arbitrarily large condition numbers and badly behaved data in some online rounds may breakdown the restricted strong convexity. Moreover, they focused on the regret bound instead of parameter estimation errors or prediction errors, which are typically used in the statistics literature.

Statistical perspective: Closely related to our work, Fan et al. (2018a) proposed a two-stage algorithm that first conducts a burn-in stage that identifies the support of the sparse underlying coefficient through solving an offline LASSO problem, and conducts the online learning stage that employs a fixed number of truncated gradient descent steps onto the pre-identified support set upon receiving each online data. Unfortunately, they require to identify the true support without errors within the first burn-in stage, which further relies on a minimal signal strength assumption and a sufficient large initial data batch so that the true support can be correctly determined with high probability. Meanwhile, this approach also requires the loss functions to satisfy the restricted strong convexity condition uniformly over all online learning rounds. These assumptions are rather stringent in practice.

2 FROM OFFLINE TO ONLINE

Notation. We first summarize the notation used throughout the paper. For any positive integer K, we write $[K] = \{1, 2, \cdots, K\}$, the collection of positive integers up to K. For any two sequences of positive real numbers $\{a_n\}$ and $\{b_n\}$, we write $a_n \lesssim b_n$ if there exist a constant $C > 0$ and a positive integer n_0 such that $a_n \leq Cb_n$ for all $n \geq n_0$.

This section develops an algorithm for online sparse linear regression. We consider an online environment where data arrive sequentially, and we only have one single machine that has limited storage and memory. Assume the covariate vector $x_i \in \mathbb{R}^p$ and the response $y_i \in \mathbb{R}$ follow the linear regression model:

$$y_j = x_j^\top \beta^* + \epsilon_j,$$

where $\beta^* \in \mathbb{R}^p$ is the underlying sparse regression coefficient vector such that $\|\beta^*\|_0 \ll p$ and ϵ_j is a random noise.

To estimate the underlying coefficient β^* under the linear model, a commonly adopted loss function is the squared loss ℓ_j, which calculates the squared error for each data point (x_i, y_i) that

$$\ell_j(\beta) = \frac{1}{2}(y_j - x_j^\top \beta)^2.$$

We consider an online environment where we have access to an initial batch consisting of t_0 independently generated data points $(x_{0j}, y_{0j})_{1 \leq j \leq t_0}$. For the initial batch, with a slight abuse of notation, we define its corresponding loss function as

$$\ell_0(\beta) = \frac{1}{2t_0} \sum_{j=1}^{t_0} (y_{0j} - x_{0j}^\top \beta)^2,$$

which is the averaged loss over each data point within the initial batch. Subsequently, one data point (x_t, y_t) comes in each online learning round t.

In high-dimensional statistics, the LASSO approach has been widely employed to obtain sparse estimators. Ideally, if our machine has infinite memory and storage, after the reception of the first t data points as well as the initial batch, the offline Lasso estimates the regression coefficient vector β^* by directly solving

$$\min_{\beta \in \mathbb{R}^p} \Big\{ \frac{1}{t + t_0} \Big(t_0 \ell_0(\beta) + \sum_{j=1}^{t} \ell_j(\beta) \Big) + \lambda_t \|\beta\|_1 \Big\}. \quad (2)$$

The above LASSO formulation works well for the offline problem, but for the online scenario, it suffers from three key challenges mentioned in Section 1, namely memory and storage restriction, dynamic update of the regularization parameter, and uniform restricted strong convexity condition. Because of the these issues, developing online sparse linear regression has been particularly challenging.

To overcome the above issues, we propose a novel memory efficient framework for solving the online sparse regression problem. Our new framework replaces all the loss functions for the online data, excluding the initial batch, by their linear approximation, approximated using the current best available estimator, denoted by $\tilde{\beta}$. We assign a fixed weight to the initial batch and allows general weights for the data obtained for each online learning round. Specifically, let $w_{t,j}$ be the weight for the j-th data (x_j, y_j) and let $W_t = \sum_{j=1}^{t} w_{t,j}$ be the total weights, we define the loss function as

$$\begin{aligned} &L_t(\beta; \tilde{\beta}) \\ &= \ell_0(\beta) - \nabla \ell_0(\tilde{\beta})^\top \beta + \Big\langle \sum_{j=1}^{t} \frac{w_{t,j}}{W_t} \nabla \ell_j(\tilde{\beta}), \beta \Big\rangle, \end{aligned} \quad (3)$$

where we have singled out the initial batch and distributed the rest as linear approximations. Here we call $\tilde{\beta}$ the *root* of the loss function to evaluate the linear approximation terms. In our formulation, the quadratic term $\ell_0(\beta)$ resulted from the initial batch helps provide the curvature of the loss function, while the linear approximation terms utilizes data from the subsequent online learning rounds, which would not affect the curvature but help us improve the statistical accuracy of the obtained solution at the current online round. Note that for the initial batch where $t = 0$, we do not introduce any linear approximation and the above loss function reduces to the standard least-squared loss.

To better understand the above loss function, let us calculate its gradient at β:

$$\nabla L_t(\beta; \tilde{\beta}) = \nabla \ell_0(\beta) - \nabla \ell_0(\tilde{\beta}) + \sum_{j=1}^{t} \frac{w_{t,j}}{W_t} \nabla \ell_j(\tilde{\beta}).$$

Although the initial batch is fixed, the term $\nabla \ell_0(\beta) - \nabla \ell_0(\tilde{\beta})$ is small provided $\tilde{\beta}$ is close to β. Intuitively, if we

can improve our root estimator and find a sequence $\{\tilde{\beta}_t\}$ converging to the underlying coefficient β^*, then

$$\nabla L_t(\beta;\tilde{\beta}_t) \approx \sum_{j=1}^{t} \frac{w_{t,j}}{W_t}\nabla \ell_j(\tilde{\beta}_t) \approx \sum_{j=1}^{t} \frac{w_{t,j}}{W_t}\nabla \ell_j(\beta^*),$$

whose decay rate depends on the online data $\{(x_i, y_i)\}_{i=1}^t$ but *not* the initial batch.

As we have seen, the linear term $-\nabla\ell_0(\tilde{\beta})^\top\beta$ introduced in the loss function (3) helps us break the bottleneck induced by the initial batch $\ell_0(\beta)$ without rescaling it as $\frac{t_0}{t_0+t}\ell_0(\beta)$, which is commonly adopted by LASSO formulation (2) but would lose the curvature as the rescaling factor $\frac{t_0}{t_0+t} \to 0$ when $t \to \infty$. This makes our framework memory efficient and preserves the curvature information simultaneously.

In addition, because the curvature of our loss function is provided by the initial batch instead of online data, as we will discuss in Section 3, the loss functions in all online rounds satisfy the RSC condition uniformly once it holds for the initial batch squared-loss $\ell_0(\beta)$. This is another advantage of our new framework.

2.1 Algorithm

After introducing the loss function, we formally state our algorithm for solving streaming sparse regression. Our online Lasso algorithm consists of the following two stages. In the first stage, we calculate our initial estimator as

$$\hat{\beta}_0 = \underset{\beta\in\mathbb{R}^p}{\arg\min}\Big\{\ell_0(\beta) + \lambda_0\|\beta\|_1\Big\}. \tag{4}$$

Our second stage recursively solves a l_1-regularized optimization problem, with $\hat{\beta}_0$ being the initialization point. Specifically, letting $\hat{\beta}_{t-1}$ be the optimal solution obtained for iteration $t-1$, upon receiving the t-th data point, we adopt $L_t(\beta;\hat{\beta}_{t-1})$ as the loss function and solve

$$\hat{\beta}_t = \underset{\beta\in\mathbb{R}^p}{\arg\min}\Big\{L_t(\beta;\hat{\beta}_{t-1}) + \lambda_t\|\beta\|_1\Big\}. \tag{5}$$

We collect the algorithm in Algorithm 1 and refer to this approach as the Online Linearized LASSO (OLin-LASSO).

Note the the loss function preserves a simple structure that it is the sum of a quadratic function and a linear term, which can be efficiently solved by various algorithms, such as FISTA (Beck and Teboulle, 2009).

2.2 Memory Efficient Weighting Scheme

Before proceeding, let us briefly discuss the memory cost and updating schemes under our framework. For the squared loss, our proposed algorithm only requires minimal memory and storage space, independent of t. We discuss this by considering two cases in the following.

Algorithm 1 Online Linearized (OLin) LASSO

Require: $\{\lambda_t\}$, initial batch size t_0, online rounds T
1: generate t_0 samples,
2: compute $\hat{\beta}_0$ by (4).
3: **for** $j = 1, \cdots, t$ **do**
4: receive the data pair (x_j, y_j).
5: compute $\hat{\beta}_j$ by (5).
6: **end for**
7: **return** $\{\hat{\beta}_j\}_{j=1}^t$

t-independent weights: For t-independent weights $w_{t,j} = w_j$, the proposed algorithm only needs $\mathcal{O}(t_0p + p^2)$ memory space to record summary statistics from the history, which is independent of t. Note that

$$\sum_{j=1}^{t} \frac{w_{t,j}}{W_t}\nabla\ell_j(\beta) = \sum_{j=1}^{t} \frac{w_{t,j}}{W_t}(x_jx_j^\top\beta - x_jy_j).$$

Therefore, we could store the terms $s_t = \sum_{j=1}^t w_{t,j}x_jx_j^\top$ and $r_t = \sum_{j=1}^t w_{t,j}x_jy_j$ to evaluate the gradient. Specifically, upon receiving the data point (x_t, y_t), to evaluate the cost function, it suffices to record

$$s_t = s_{t-1} + w_tx_tx_t^\top, r_t = r_{t-1} + w_tx_ty_t.$$

Then in the t-th step, the machine calculates

$$\sum_{j=1}^{t} \frac{w_{t,j}}{W_t}\nabla\ell_j(\hat{\beta}_{t-1}) = \frac{1}{W_t}\Big(s_t\hat{\beta}_{t-1} - r_t\Big).$$

t-dependent weights: When $w_{t,j}$'s are t-dependent such that $w_{t,j} = 1/t$, we also only need $\mathcal{O}(pt_0 + p^2)$ memory space to record the historical data. For any given β, to evaluate the cost function, it suffices to record

$$s_t = s_{t-1} + x_tx_t^\top, r_t = r_{t-1} + x_ty_t.$$

Then in the t-th step, the algorithm calculates

$$\sum_{j=1}^{t} \frac{w_{t,j}}{W_t}\nabla\ell_j(\hat{\beta}_{t-1}) = \frac{1}{t}\Big(s_t\hat{\beta}_{t-1} - r_t\Big).$$

In both cases, the total memory space needed for the $(t+1)$-step is $\mathcal{O}(t_0p + p^2)$, independent of t.

3 THEORY

After introducing our OLin-LASSO Algorithm 1, we are now ready to study the statistical properties of the solution sequence $\{\hat{\beta}_t\}$ generated by our algorithm.

To facilitate the analysis, we first introduce a restricted cone around the sparse underlying coefficient β^*. Specifically,

for any set of entry indices $\mathcal{A} \subset [p]$, we define the following restricted cone

$$\mathcal{C}_{\mathcal{A}} := \Big\{\xi \in \mathbb{R}^p : \|\xi_{\mathcal{A}^c}\|_1 \leq 3\|\xi_{\mathcal{A}}\|_1\Big\}, \tag{6}$$

where $\xi_{\mathcal{A}}$ is the vector with entries within $\mathcal{A}$ being the same as those in ξ and the rest being zeros. That is, $[\xi_{\mathcal{A}}]_i = \xi_i$ if $i \in \mathcal{A}$ and $[\xi_{\mathcal{A}}]_i = 0$ otherwise. To analyze the sparsity property of our estimator $\hat{\beta}_t$, we utilize the above restricted cone and show that when the regularization coefficient λ_t is properly chosen, $\hat{\beta}_t - \beta^*$ falls within the restricted cone induced by the true support S as follows.

Lemma 3.1. If $\lambda_t \geq 2\|\nabla L_t(\beta^*; \hat{\beta}_{t-1})\|_\infty$, then $\hat{\beta}_t - \beta^* \in \mathcal{C}_S$, where $S = \text{support}(\beta^*)$ and $\mathcal{C}_S$ is the cone defined in (6).

We defer the detailed proof to Section A.1 of the supplement.

The above result is deterministic and holds regardless of the distribution of the covariate vector x_i and response y_i. It implies that by choosing a sufficiently large regularizer λ_t, problem (5) generates an approximately sparse estimator such that $\hat{\beta}_t - \beta^*$ falls within the restricted cone $\mathcal{C}_S$.

We set $\lambda_t \propto 2\|\nabla L_t(\beta^*; \hat{\beta}_{t-1})\|_\infty$ in the rest of this paper. To further study the performance of our estimators, we assume the squared loss induced by the initial batch satisfies the following RSC condition.

Assumption 3.2. A function $f(\beta)$ is said to satisfy the RSC condition if there exists $\kappa > 0$ such that for any $\Delta \in \mathcal{C}_S$,

$$f(\beta^* + \Delta) - f(\beta^*) - \Delta^\top \nabla f(\beta^*) \geq \kappa\|\Delta\|_2^2,$$

where $S = \text{support}(\beta^*)$ and $\mathcal{C}_S$ is the cone defined in (6).

Raskutti et al. (2010) have shown that the above RSC condition holds provided $n \geq Cs\log(p)$ under sub-Gaussian designs. Furthermore, as our loss function $L_t(\bullet; \tilde{\beta})$ consists of a squared loss $\ell_0(\beta)$ and a linearized term, if $\ell_0(\beta)$ satisfies the RSC condition, then the RSC condition also holds for all $L_t(\bullet; \tilde{\beta})$, which is formally stated as follows.

Proposition 3.3. Suppose $\ell_0(\beta)$ satisfies the RSC condition with parameter κ, then for any $t \geq 1$, $\nabla L_t(\bullet; \tilde{\beta})$ also satisifies the RSC condition with parameter κ.

We emphasize that under our framework, the RSC condition holds for *all* online learning rounds once it holds for the initial batch. As a result, our framework only requires to verify the RSC condition *once* in the initialization phase. In contrast, naively solving a standard LASSO-based problem in each online round would require to verify the RSC condition in all rounds, which is quite stringent. This restricts its applicability to real-world problems.

The RSC property of our loss function allows us to further understand the behaviors of estimators $\hat{\beta}_t$. In what follows, we build up an upper bound for $\|\hat{\beta}_t - \beta^*\|_1$ in terms of the regularization cofficient λ_t and ℓ_∞-norm $\|\nabla L_t(\beta^*; \hat{\beta}_{t-1})\|_\infty$ of the gradient evaluated at the underlying coefficient β^*.

Lemma 3.4. Suppose $\ell_0(\beta)$ satisfies the RSC Assumption 3.2 and $\lambda_t \geq 2\|\nabla L_t(\beta^*; \hat{\beta}_{t-1})\|_\infty$, then

$$\|\hat{\beta}_t - \beta^*\|_1 \leq \frac{16s}{\kappa}\big(\lambda_t + \|\nabla L_t(\beta^*; \hat{\beta}_{t-1})\|_\infty\big) \leq \frac{24s}{\kappa}\lambda_t$$

$$\text{and } \|\hat{\beta}_t - \beta^*\|_2^2 \leq \frac{\lambda_t + \|\nabla L_t(\beta^*; \hat{\beta}_{t-1})\|_\infty}{\kappa}\|\hat{\beta}_t - \beta^*\|_1.$$

We defer the detailed proof to Section A.2 of the supplement.

To further quantify the above statistical error, it suffices to provide a statistical bound for $\|\nabla L_t(\beta^*; \hat{\beta}_{t-1})\|_\infty$. We start with writing

$$\begin{aligned}&\nabla L_t(\beta^*; \hat{\beta}_{t-1})\\ &= \sum_{j=1}^{t} \frac{w_{t,j}}{W_t}\nabla \ell_j(\beta^*) + \Big(\hat{\Lambda}_t - \hat{\Lambda}_0\Big)(\hat{\beta}_{t-1} - \beta^*),\end{aligned} \tag{7}$$

where $\hat{\Lambda}_t = \frac{1}{W_t}\sum_{j=1}^{t} w_{t,j}x_j x_j$ and $\hat{\Lambda}_0 = \frac{1}{t_0}\sum_{i=1}^{t_0} x_i x_i$ represent the weighted average of $x_i x_i^\top$'s collected in the online phase and initial batch, respectively.

Letting $\Lambda = \mathbb{E}[x_i x_i^\top]$, we decompose $\hat{\Lambda}_t - \hat{\Lambda}_0$ as

$$\hat{\Lambda}_t - \hat{\Lambda}_0 = \sum_{j=1}^{t} \frac{w_{t,j}}{W_t}(x_j x_j - \Lambda) + \frac{1}{t_0}\sum_{i=1}^{t_0}(\Lambda - x_i x_i).$$

With such a decomposition, each entry of $\frac{w_{t,j}}{W_t}(x_j x_j - \Lambda)$ is mean-zero so that their sum can be viewed as a martingale. Meanwhile, under mild tail assumptions of the data, both of the above terms concentrate with t_0 and the online learning rounds t increasing.

For now, we impose the following assumption on the covariate x_i and noise ϵ_i.

Assumption 3.5. Each covariate $x_i \in \mathbb{R}^p$ is independently and identically distributed such that $x_i \sim$ sub-Gaussian(σ_X^2).[1] Each noise ϵ_i is independently generated and follows a sub-Gaussian distribution such that $\epsilon_i \sim$ sub-Gaussian(σ_ϵ^2) for some $\sigma_\epsilon > 0$.

The above sub-Gaussian assumption is mild and widely adopted by the high-dimensional statistics literature; see for example Raskutti et al. (2010); Wainwright (2019).

To analyze the concentration property of $\hat{\Lambda}_t - \hat{\Lambda}_0$, one major difficulty is that the nonasymptotic upper bound must hold uniformly over all rounds $j = 1, 2, \cdots, t$, so we can

[1] A zero-mean random vector X is said to be sub-Gaussian(σ^2) if for each fixed unit vector $\mathbb{E}[e^{\lambda\langle u, X\rangle}] \leq e^{\lambda^2\sigma^2/2}$ for all $\lambda \in \mathbb{R}$.

utilize this to derive an upper error bound for $\|\hat{\beta}_t - \beta^*\|_1$ that holds uniformly for the entire solution trajectory.

In order to do so, we need the following lemma for bounding the probability of an upper tail of a sub-martingale, whose proof is provided in Section A.3.

Lemma 3.6. Assume (X_k) is a sequence of independent sub-Gaussian random variables, with each X_k having the sub-Gaussian norm given by σ_k such that $\mathbb{E}[\exp(\nu X_k)] \leq \exp(\frac{\sigma_k^2 \nu^2}{2})$ for all $\nu > 0$. If we define $S_n = \sum_{k=1}^n X_k$, then

$$\mathbb{P}(\max_{1\leq i\leq n} S_i \geq t) \leq \exp\left(-\frac{t^2}{2\sum_{i=1}^n \sigma_i^2}\right).$$

In the rest of this paper, we write $z_j = \frac{\sqrt{\sum_{k=1}^j w_k^2}}{\sum_{k=1}^j w_k}$ for notational convenience. The following result characterizes the concentration properties of $\hat{\Lambda}_j - \Lambda$ and $\sum_{j=1}^t \frac{w_{t,j}}{W_t} \nabla \ell_j(\beta^*)$.

Proposition 3.7. Suppose the weights are

$$w_{t,j} = \frac{1}{j^a}, z_j = \frac{\sqrt{\sum_{k=1}^j w_k^2}}{\sum_{k=1}^j w_k}.$$

for some $0 \leq a < 1$ and Assumption 3.5 holds. For any $\epsilon > 0$, there exists a constant $c > 0$ such that with probability at least $1 - 4\delta$, for all $j = 1, 2, \cdots, t$,

$$\|\hat{\Lambda}_0 - \Lambda\|_{\max} \leq c t_0^{-1/2} \sqrt{\log(p^2/\delta)},$$
$$\|\hat{\Lambda}_j - \Lambda\|_{\max} \leq c z_j \sqrt{\log(p^2/\delta)},$$
$$\|\sum_{k=1}^j \frac{w_{j,k}}{W_j} \nabla \ell_k(\beta^*)\|_\infty \leq c z_j \sqrt{\log(pt/\delta)} \sqrt{\log(p/\delta)}.$$

By applying the above uniform martingale bounds to the ℓ_∞ of $\nabla L_t(\beta^*; \hat{\beta}_{t-1})$ in (7), we obtain the following result.

Proposition 3.8. Suppose Assumption 3.5 holds. For all $\epsilon > 0$, there exists a universal constant $c_\epsilon > 0$ such that with probability at least $1 - 4\delta$, for all $j \leq t$,

$$\begin{aligned}&\|\nabla L_j(\beta^*; \hat{\beta}_{j-1})\|_\infty \\ &\leq c z_j \sqrt{\log(pt/\delta)} \sqrt{\log(p/\delta)} \\ &\quad + 2c\sqrt{\log(p^2/\delta)}\Big(z_j + \frac{1}{t_0^{1/2}}\Big)\|\hat{\beta}_{j-1} - \beta^*\|_1.\end{aligned}$$

Now, we consider the scenario where the weights $w_{t,k}$ are t-independent. Thus from now on, we sometimes write $w_{t,k}$ as w_k. Recall the definition of z_t in Proposition 3.7. We first quantify z_t as follows.

Corollary 3.9. Suppose the weight are t-independent and set in the form that $w_{t,j} = \frac{1}{j^a}$ for $0 \leq a < 1$, then there exists a constant $c_0 > 0$ such that for all $j \leq t$,

(i) $z_j \leq \frac{c_0}{\sqrt{j}}$ if $0 \leq a < 1/2$;

(ii) $z_j \leq \frac{c_0 \ln j}{\sqrt{j}}$ if $a = 1/2$;

(iii) $z_j \leq \frac{c_0}{j^{1-a}}$ if $\frac{1}{2} < a < 1$.

Here we note that the decay rate of $\|\nabla L_t(\beta^*; \hat{\beta}_{t-1})\|_\infty$ is determined by the diminishing rate of z_t. Corollary 3.9 indicates that z_t achieves the fastest diminishing rate when $0 \leq a < 1/2$. Thus, in the rest of our paper, we adopt this choice of weights so that $z_j \lesssim j^{-1/2}$. Next, by combining Proposition 3.8 with Lemma 3.4 and setting $\lambda_j \propto 2\|\nabla L_j(\beta; \hat{\beta}_{j-1})\|_\infty$, for all $j \leq t$, we can bound the ℓ_1 error as

$$\begin{aligned}\|\hat{\beta}_j - \beta^*\|_1 \leq \frac{48sc}{\kappa}\sqrt{\frac{\log(pt/\delta)}{j}}\log(p/\delta) \\ + \frac{96sc}{\kappa}\Big(\frac{c_0}{j^{1/2}} + \frac{1}{t_0^{1/2}}\Big)\|\hat{\beta}_{j-1} - \beta^*\|_1.\end{aligned}$$

The above result suggests that with high probability, for each online learning round j, the statistical error $\|\hat{\beta}_j - \beta^*\|_1$ can be bounded in terms of the statistical error $\|\hat{\beta}_{j-1} - \beta^*\|_1$ incurred within the previous online learning round plus an $\mathcal{O}(\sqrt{\frac{\log t}{j}})$ term, which serves as the building block in analyzing the behavior of $\{\hat{\beta}_j\}_{j=1}^t$.

Finally, because the above relationship holds for all $j \leq t$, by recursively applying it, we characterize the convergence behavior of $\|\hat{\beta}_t - \beta^*\|_1$ as follows, and provide its proof in Section A.4.

Theorem 3.10. Suppose Assumption 3.5 holds and $\ell_0(\beta)$ satisfies the RSC condition (3.2). With probability at least $1 - 4\delta$, we have

$$\begin{aligned}&\|\hat{\beta}_t - \beta^*\|_1 \\ &\leq \sum_{j=1}^t \left(\prod_{k=j+1}^t a_k\right) b_j + \left(\prod_{j=1}^t a_j\right)\|\hat{\beta}_0 - \beta^*\|_1,\end{aligned}$$

where

$$a_j = \frac{96sc}{\kappa}\sqrt{\log(p^2/\delta)}\left(\frac{c_0}{j^{1/2}} + \frac{1}{t_0^{1/2}}\right),$$
$$b_j = \frac{48sc}{\kappa}\sqrt{\frac{\log(pt/\delta)}{j}}\sqrt{\log(p/\delta)}.$$

In addition, suppose the size of initial batch t_0 ensures $a_j < 1$ for large j, we have with probability at least $1 - 4\delta$,

$$\|\hat{\beta}_t - \beta^*\|_1 \leq \widetilde{\mathcal{O}}\Big(\frac{s}{\kappa\sqrt{t}}\Big) \text{ and } \|\hat{\beta}_t - \beta^*\|_2 \leq \widetilde{\mathcal{O}}\Big(\frac{\sqrt{s}}{\kappa\sqrt{t}}\Big).$$

Remark 3.11. It is worth pointing out that the convergence of our algorithm is ensured if the size of initial batch $t_0 \geq (96sc/\kappa)^2 \log(p^2/\delta)$, which only depends on the distribution of covariate x_j's, the condition number κ, and the

sparsity level $s = \|\beta^*\|_0$, instead of the underlying parameter β^*. In comparison with the minimal signal assumption $\min_{j\in S} |\beta_j^*| \geq 2c\sqrt{\log p/t_0}$ by Fan et al. (2018a), which guarantees the support of β^* can be identified by the initial batch with t_0 data points, our approach completely removes this minimal signal assumption. Consequently, our approach would exhibit superior performance in weak signal scenarios where $\min_{j\in S} |\beta_j^*|$ is small or the initial batch is small. We verify this phenonmenon in our numerical experiments.

Remark 3.12. Notably, as discussed in Proposition 3.3, all loss functions $\{L_j(\bullet; \tilde{\beta})\}_{j=0}^t$ satisfy the RSC condition once it holds for the squared loss $\ell_0(\beta)$ induced by the initial batch. Therefore, when employing our framework, it suffices to guarantee that the RSC condition *once* for the initial batch instead of guaranteeing it for all online rounds. This boosts the applicability of our method in practice especially when there exist badly behaved data points in certain rounds.

Remark 3.13. Theorem 3.10 quantifies the behavior of the entire solution trajectory $\{\hat{\beta}_j\}_{j=1}^t$ and ensures its good performance over all online rounds. This suggests that our approach produces better and better estimators with the online learning round increasing. The solution sequence converges to the underlying sparse coefficient vector β^* at the rate of $\widetilde{\mathcal{O}}(\sqrt{s/t})$ in terms of the ℓ_2-norm error, which matches the minimax lower bound of the sparse linear regression in the offline case up to logarithmic terms (Raskutti et al., 2011). Compared with existing work on offline ℓ_2-regularized regression that requires to store the entire dataset of cost $\mathcal{O}(pt)$, such as the offline LASSO, our approach enjoys a much lower *fixed* memory cost of $\mathcal{O}(pt_0 + p^2)$ and is favored for high-dimension big-data applications where both covariate dimension p and number of data points t are large.

4 NUMERICAL EXPERIMENTS

After studying the theoretical performance of our `Olin_LASSO` algorithm in Section 3, we now continue to investigate the practical performance of our algorithm in various settings. We consider the scenario where the covariate vector is of high dimensionality such that $p = 1000$. We consider the *weak* signal scenario and generate the underlying coefficient β^* as follows: We generate a sparse vector whose first 20 entries are nonzero and the rest are zeros. In this case, we let $S = \text{Support}(\beta^*) = \{1, 2, \cdots, 20\}$ be the support of underlying β^* and have the sparsity level $s = |S| = \|\beta^*\|_0 = 20$. For each entry $j \in S$, we independently generate a weak signal such that $\beta_j^* \sim \mathcal{N}(0, 0.25)$. In our experiments, the mean absolute value of the these nonzero entries is 0.261, and the ℓ_1- and ℓ_2-norm of β^* are $\|\beta^*\|_1 = 5.22$ and $\|\beta^*\|_2 = 1.30$, respectively.

We investigate the numerical performance of our algorithm against a baseline algorithm `OS_LASSO_K` (Fan et al., 2018a). `OS_LASSO_K` is a two-phase algorithm that adopts the standard least-squared loss $h_t(\beta)$ instead of our variant. In the burn-in phase, `OS_LASSO_K` determines the set of nonzero entries S_0 of β^* by running a penalized regression using the initial batch of size t_0. We will use LASSO in all of our experiments. In the online learning phase, upon receiving each online data point (x_i, y_t), `OS_LASSO_K` conducts K iterative hard-thresholding gradient descent steps $\beta_{i,k+1} = \Pi_{S_0}(\beta_{i,k} - \eta\nabla h_t(\beta_{i,k}))$ for $k = 1, \cdots, K$ that only keeps the entries within S_0 and truncates the rest entries to zero.

We test our algorithm and the baseline algorithms over the covariate correlation setup:

- Toeplitz $\rho = 0.5$ correlation: We generate the covariate $x_i \in \mathbb{R}^p$ under a multivariate norm distribution that $x_i \sim \mathcal{N}(\mathbf{0}, \Sigma)$ where $\Sigma \in \mathbb{R}^{p\times p}$ is a covariance matrix such that $\Sigma_{i,i} = 1$ and $\Sigma_{i,j} = \rho^{|i-j|}$ for all $i \neq j$.

In the above setup, to generate data pair (x_i, y_i), we first generate the covariate $x_i \in \mathbb{R}^p$ as above, then independently generate a noise term $\epsilon_i \sim \mathcal{N}(0, 1)$, and set the response as $y_i = x_i^\top \beta^* + \epsilon_i$.

We consider following two experiments:

1. We test the performance of above algorithms over different initial batch sizes $t_0 = 50, 100, 150, \cdots, 500$ under the above Toeplitz $\rho = 0.5$ correlation design. After running each simulation for $T = 10^4$ online learning rounds, we plot the obtained mean squared error (MSE) $\|\beta_T - \beta^*\|_2^2$ against the initial batch size t_0 in Figure 1. We set the regularization coefficient for our `Olin_LASSO` algorithm as $\lambda_t = \sqrt{\frac{\log(p)}{t}}$, and set the step-size of the baseline algorithms `OS_LASSO_K=1` and `OS_LASSO_K=20` as $\eta = 0.001$.
2. We fix the initial batch size as $t_0 = 100$, test the algorithms over $T = 10^4$ online rounds under the above covariate design, and plot the MSE $\|\beta_t - \beta^*\|_2^2$ against online round t in Figure 2. Other parameters are set as the same as the first experiment.

From Figure 1, we observe that our algorithm outperforms `OS_LASSO_K=1` and `OS_LASSO_K=20`, especially when the initial batch size t_0 is small. For example, when $t_0 = 100$, `OS_LASSO_K=1` and `OS_LASSO_K=20` generates estimator of MSE being around 0.49, while our `Olin_LASSO` algorithm generate a better estimator that has a MSE around 0.05.

Further, from Figure 2, we can see that `OS_LASSO_K=1` and `OS_LASSO_K=20` do not generate sequences that converge to the underlying estimator β^*, as evidenced by the

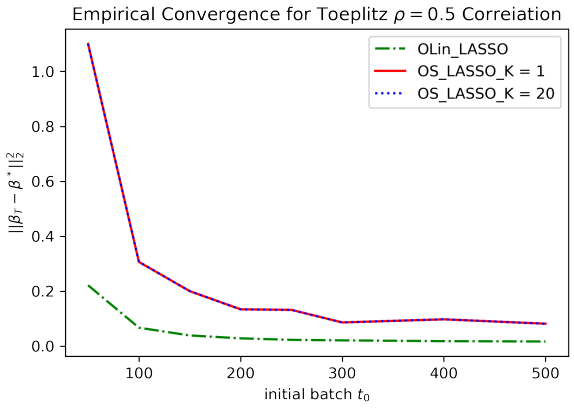


Figure 1: Empirical Convergence of MSE $\|\beta_T - \beta^*\|_2^2$ under weak signal setup for $t_0 = 50, 100, 150, \cdots, 500$ and $T = 10000$ online learning rounds for Toeplitz $\rho = 0.5$ covariate correlation design.

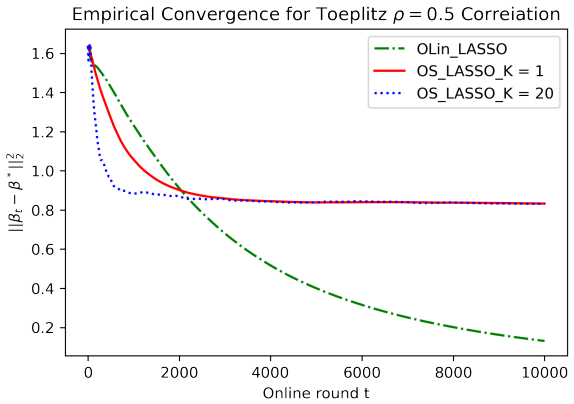


Figure 2: Empirical Convergence of MSE $\|\beta_t - \beta^*\|_2^2$ under weak signal setup for online learning rounds $t \in [0, 10000]$ and initial batch size $t_0 = 100$ for Toeplitz $\rho = 0.5$ covariate correlation design.

observation that the MSE is not improving when receiving more online data. In contrast, our algorithm can generate consistent estimators that converge to the underlying β^*.

Both of the above experiments suggest that `OS_LASSO` is sensitive to the size of initial batch t_0 and can only converge when t_0 is large enough to determine the support of β^*, which requires a large initial batch in the weak signal scenario. By contrast, our algorithm exhibits superior performance to `OS_LASSO` even if only a small initial batch is available.

To verify the rate of convergence provided by our theorem, we plot the log-error $\log(\|\beta_t - \beta^*\|_1)$ against the log-round $\log t$ in Figure 3. We observe that it preserves a slope around $-1/2$. This suggests that $\{\hat{\beta}_t\}$ indeed converges to β^* at the rate of $\tilde{\mathcal{O}}(\frac{1}{\sqrt{T}})$ under ℓ_1-norm, which matches Theorem 3.10.

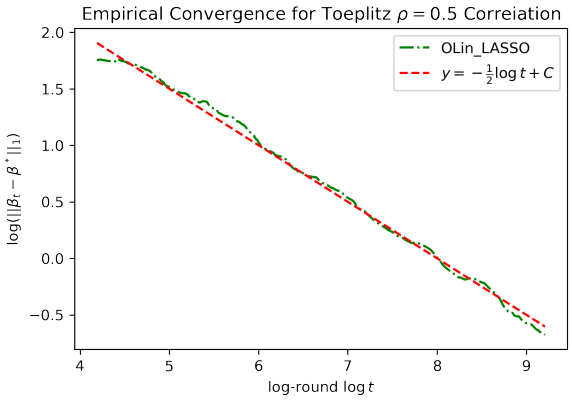


Figure 3: $\log(\|\beta_t - \beta^*\|_1)$ against $\log(t)$ under weak signal setup for $t_0 = 500$ and $T = 10000$ online learning rounds for Toeplitz $\rho = 0.5$ covariate correlation design.

In addition, we test our algorithm against the baselines under other types of covariate correlation distributions. We also test these algorithms under a strong signal setting where β^* has significant absolute values for nonzero entries. Our algorithm exhibits superior performance against the baseline algorithms in all scenarios. We provide detailed experiment setups and numerical results in Section B of the supplement. We also conduct experiments on heavy-tailed data and small initial batches $t_0 \in \{1, 10\}$ and provide results in Section B. The detailed codes are provided at `github.com/shuoguangyang/OLinLASSO`.

5 CONCLUSION

In this paper, we propose a novel framework to solve the online sparse linear regression problem. Our framework adopts a loss function that consists of a squared loss induced by the initial batch and a linearized term induced by the online data, and our approach is memory efficient with a memory complexity of $\mathcal{O}(pt_0 + p^2)$. By ensuring the RSC condition only for the initial squared loss, the entire solution trajectory $\{\hat{\beta}_t\}$ falls into a restricted set, and converges to β^* in the optimal rate of $\widetilde{\mathcal{O}}(\sqrt{s/t})$ under ℓ_2-norm, establishing the benchmark in online sparse linear regression.

In addition, our algorithm exhibits superior practical performance comparing with existing arts under various design correlations. These numerical experiments also aligns with our nonasymptotic rate of convergence result well.

One possible limitation of our framework is that it requires an initial batch of size $t_0 \geq (96sc/\kappa)^2 \log(p^2/\delta)$ to ensure convergence, where the primitives such as the sparsity level s and the conditional number κ are unknown in practice. It remains an open problem whether problem-independent algorithms can be developed to avoid the usage of such primitives but still exhibit a comparable performance theoretically.

Acknowledgements

The work described in this paper was partially supported by the Early Career Scheme from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. 26209422 to Shuoguang Yang) and Natural Sciences and Engineering Research Council of Canada (grant RGPIN-2018-06484 to Qiang Sun).

References

Agarwal, A., Negahban, S., and Wainwright, M. J. (2012). Fast global convergence rates of gradient methods for high-dimensional statistical recovery. *The Annals of Statistics*, 40(5):2452–2482.

Beck, A. and Teboulle, M. (2009). A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences*, 2(1):183–202.

Bertsekas, D. P. (2011). Incremental proximal methods for large scale convex optimization. *Mathematical programming*, 129(2):163–195.

Duchi, J., Hazan, E., and Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(7).

Fan, J., Gong, W., Li, C. J., and Sun, Q. (2018a). Statistical sparse online regression: A diffusion approximation perspective. In *International Conference on Artificial Intelligence and Statistics*, pages 1017–1026. PMLR.

Fan, J., Liu, H., Sun, Q., and Zhang, T. (2018b). I-lamm for sparse learning: Simultaneous control of algorithmic complexity and statistical error. *Annals of statistics*, 46(2):814.

Foster, D., Karloff, H., and Thaler, J. (2015). Variable selection is hard. In *Conference on Learning Theory*, pages 696–709. PMLR.

Garrigues, P. and Ghaoui, L. (2008). An homotopy algorithm for the lasso with online observations. *Advances in neural information processing systems*, 21.

Kale, S. (2014). Open problem: Efficient online sparse regression. In *Conference on Learning Theory*, pages 1299–1301. PMLR.

Kale, S., Karnin, Z., Liang, T., and Pál, D. (2017). Adaptive feature selection: Computationally efficient online sparse linear regression under rip. In *International Conference on Machine Learning*, pages 1780–1788. PMLR.

Kushner, H. and Yin, G. (1997). *Stochastic Approximation Algorithms and Applications*. Springer-Verlag.

Langford, J., Li, L., and Zhang, T. (2009). Sparse online learning via truncated gradient. *Journal of Machine Learning Research*, 10(3).

Loh, P.-L. and Wainwright, M. J. (2015). Regularized m-estimators with nonconvexity: statistical and algorithmic theory for local optima. *Journal of Machine Learning Research*, 16:559–616.

Rakhlin, A., Shamir, O., and Sridharan, K. (2011). Making gradient descent optimal for strongly convex stochastic optimization. *arXiv preprint arXiv:1109.5647*.

Raskutti, G., Wainwright, M. J., and Yu, B. (2010). Restricted eigenvalue properties for correlated gaussian designs. *The Journal of Machine Learning Research*, 11:2241–2259.

Raskutti, G., Wainwright, M. J., and Yu, B. (2011). Minimax rates of estimation for high-dimensional linear regression over l_q-balls. *IEEE transactions on information theory*, 57(10):6976–6994.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, 58:267–288.

Wainwright, M. J. (2019). *High-dimensional statistics: A non-asymptotic viewpoint*, volume 48. Cambridge University Press.

Xiao, L. (2009). Dual averaging method for regularized stochastic learning and online optimization. *Advances in Neural Information Processing Systems*, 22.

Yang, H., Xu, Z., King, I., and Lyu, M. R. (2010). Online learning for group lasso. In *ICML*.

A PROOF OF RESULTS

A.1 Proof of Lemma 3.1

Proof. Recall that

$$\hat{\beta}_t \in \arg\min_{\beta \in \mathbb{R}^p} \{L_t(\beta; \hat{\beta}_{t-1}) + \lambda_t \|\beta\|_1\},$$

by using the convexity of $L_t(\beta; \hat{\beta}_{t-1})$ in β, we have

$$(\hat{\beta}_t - \beta^*)^\top \nabla L_t(\beta^*; \hat{\beta}_{t-1}) \leq L_t(\hat{\beta}_t; \hat{\beta}_{t-1}) - L_t(\beta^*; \hat{\beta}_{t-1}) \leq \lambda_t(\|\beta^*\|_1 - \|\hat{\beta}_t\|_1),$$

which further yields that

$$-\|\hat{\beta}_t - \beta^*\|_1 \|\nabla L_t(\beta^*; \hat{\beta}_{t-1})\|_\infty \leq \lambda_t(\|\beta^*\|_1 - \|\hat{\beta}_t\|_1). \tag{8}$$

Because S is the support of β^*, using the inequalities

$$\|\beta_{S^c}\|_1 = \|\beta - \beta^*\|_1 - \|(\beta - \beta^*)_S\|_1$$

and

$$\|\beta^*\|_1 - \|\beta\|_1 = \|\beta^*\|_1 - \|\beta_S\|_1 - \|\beta_{S^c}\|_1 \leq \|(\beta^* - \beta)_S\|_1 - \|\beta^*_{S^c}\|_1,$$

we obtain

$$\|\beta^*\|_1 - \|\hat{\beta}\|_1 \leq 2\|(\hat{\beta} - \beta^*)_S\|_1 - \|\hat{\beta} - \beta^*\|_1.$$

By combining the above inequality with (8) and rearranging the terms, we conclude that

$$\begin{aligned}
&\frac{2\lambda_t}{\lambda_t - \|\nabla L_t(\beta^*; \hat{\beta}_{t-1})\|_\infty} \|(\hat{\beta}_t - \beta^*)_S\|_1 \\
&\quad = \frac{2}{1 - \|\nabla L_t(\beta^*; \hat{\beta}_{t-1})\|_\infty / \lambda_t} \|(\hat{\beta}_t - \beta^*)_S\|_1 \geq \|\hat{\beta}_t - \beta^*\|_1.
\end{aligned}$$

Our choice of $\lambda_t \geq 2\|\nabla L_t(\beta^*; \hat{\beta}_{t-1})\|_\infty$ makes the constant multiplier in the left-hand side upper bounded by 4, which proves the desired result. □

A.2 Proof of Lemma 3.4

Proof. Setting $\lambda_t \geq 2\|\nabla L_t(\beta^*; \hat{\beta}_{t-1})\|_\infty$, we have $\hat{\beta}_t \in \mathcal{C}_S$, which together with the RSC Assumption 3.2 implies that

$$\begin{aligned}
\lambda_t(\|\beta^*\|_1 - \|\hat{\beta}_t\|_1) &\geq \langle \nabla L_t(\beta^*; \hat{\beta}_{t-1}), \hat{\beta}_t - \beta^* \rangle + \kappa \|\hat{\beta}_t - \beta^*\|_2^2 \\
&\geq -\|\nabla L_t(\beta^*; \hat{\beta}_{t-1})\|_\infty \|\hat{\beta}_t - \beta^*\|_1 + \kappa \|\hat{\beta}_t - \beta^*\|_2^2.
\end{aligned}$$

By using the fact that $\|\hat{\beta}_t - \beta^*\|_1 \geq \|\beta^*\|_1 - \|\hat{\beta}_t\|_1$ and rearranging the above terms, we further obtain

$$\|\hat{\beta}_t - \beta^*\|_2^2 \leq \frac{\lambda_t + \| \nabla L_t(\beta^*; \hat{\beta}_{t-1})\|_\infty}{\kappa} \|\hat{\beta}_t - \beta^*\|_1.$$

Because $\hat{\beta}_t \in \mathcal{C}_S = \left\{\beta \in \mathbb{R}^p : \|(\beta - \beta^*)_{S^c}\|_1 \leq 3\|(\beta - \beta^*)_S\|_1\right\}$, we obtain

$$\|\hat{\beta}_t - \beta^*\|_1 \leq 4\|(\hat{\beta}_t - \beta^*)_S\|_1 \leq 4\sqrt{s}\|(\hat{\beta}_t - \beta^*)_S\|_2,$$

which further leads to

$$\|\hat{\beta}_t - \beta^*\|_1^2 \leq 16s\|(\hat{\beta}_t - \beta^*)_S\|_2^2 \leq \frac{16s}{\kappa}\Big(\lambda_t + \|\nabla L_t(\beta^*; \hat{\beta}_{t-1})\|_\infty\Big)\|\hat{\beta}_t - \beta^*\|_1.$$

The desired result can be acquired by dividing both sides by $\|\hat{\beta}_t - \beta^*\|_1$ and setting $\lambda_t \geq 2\|\nabla L_t(\beta^*; \hat{\beta}_{t-1})\|_\infty$. □

A.3 Proofs of Lemma 3.6, Proposition 3.7, and Proposition 3.8

For simplicity, we say a random variable X has a sub-Gaussian norm bounded by $\sigma > 0$ if $X \sim$ sub-Gaussian(σ^2).

Proof of Lemma 3.6. On one hand, the Doob's inequality implies

$$\mathbb{P}(\max_{1\le i\le n} S_i \ge t) = \mathbb{P}(\max_{1\le i\le n} e^{hS_i} \ge e^{ht}) \le \mathbb{E}[e^{hS_n - ht}] = e^{-ht}\prod_{i=1}^{n}\mathbb{E}[e^{hX_i}],$$

because e^{hS_i} is a sub-martingale. On the other hand, since all the X_i's are sub-Gaussian, we have

$$\mathbb{E}[e^{hX_i}] \le e^{h^2\sigma_i^2/2}.$$

Then we obtain

$$\mathbb{P}(\max_{1\le i\le n} S_i \ge t) \le e^{\frac{h^2}{2}\sum_{i=1}^{n}\sigma_i^2 - ht}.$$

We obtain the desired result by setting $h = (\sum_{i=1}^{n}\sigma_i^2)^{-1}t$ in the above inequality. □

Proof of Proposition 3.7. Obviously we have $\nabla\ell_k(\beta^*) = 2x_k\epsilon_k$, which means

$$S_j = \sum_{k=1}^{j} w_k\nabla\ell_k(\beta^*) = 2\sum_{k=1}^{j} w_k x_k\epsilon_k$$

is a martingale (for each component). Clearly, the sub-Gaussian norm of each component of $w_k x_k\epsilon_k$ is $w_k M_t\sigma_\epsilon$, where $M_t = \max_{i\le t, j\le p}|x_{ij}|$ and σ_ϵ represents the sub-Gaussian norm of ϵ_k. If we apply Lemma 3.6 and use a union bound, we obtain that for all $a > 0$,

$$\mathbb{P}\left(\max_{1\le j\le t}\|S_j\|_\infty \ge a\right) \le p\exp\left(-\frac{a^2}{2\sigma_\epsilon^2 M_t^2\sum_{j=1}^{t} w_j^2}\right).$$

This means that with probability at least $1-\delta$, for all $j = 1, 2, \cdots, t$, we have

$$\|S_j\|_\infty \le \sigma_\epsilon M_t\sqrt{\sum_{k=1}^{j} w_k^2}\sqrt{\log(p/\delta)}.$$

Dividing both sides by $\sum_{k=1}^{j} w_k$, we obtain that

$$\|\frac{S_j}{W_j}\|_\infty \le \sigma_\epsilon M_t\frac{\sqrt{\sum_{k=1}^{j} w_k^2}}{\sum_{k=1}^{j} w_k}\sqrt{\log(p/\delta)} = \sigma_\epsilon M_t z_j\sqrt{\log(p/\delta)}.$$

Because each $x_{i,j} \sim$ sub-Gaussian(σ_X^2), by applying the union bound, we have that for any $a > 0$,

$$\mathbb{P}\left(M_t \ge a\right) \le 2p^2\exp\left(-\frac{a^2}{2\sigma_X^2}\right).$$

Combine the above bounds, we have with probability at least $1-2\delta$, there exists a constant $c_1 > 0$ such that

$$\|\frac{S_j}{W_j}\|_\infty \le c_1 z_t\sqrt{\log(p/\delta)}\sqrt{\log(pt/\delta)}.$$

This proves the first inequality. For the second and third inequalities, note that with probability at least $1-2\delta$, there exists a constant $c_2 > 0$ satisfying that:

$$\|\hat{\Lambda}_j - \Lambda\|_\infty = \|\sum_{k=1}^{j}\frac{w_{j,k}}{W_j}(x_k x_k^\top - \Lambda)\|_\infty \le c_2 z_j\sqrt{\log(p^2/\delta)},$$

$$\text{and } \|\hat{\Lambda}_0 - \Lambda\|_\infty = \|\sum_{j=1}^{t_0}\frac{1}{t_0}(x_{0j}x_{0j}^\top - \Lambda)\|_\infty \le c_2 t_0^{-\frac{1}{2}}\sqrt{\log(p^2/\delta)}.$$

We complete the proof by setting $c = \max\{c_1, c_2\}$. □

Proof of Proposition 3.8. We first write $\nabla L_t(\beta^*; \hat{\beta}_{t-1})$ as

$$\begin{aligned}
&\nabla L_t(\beta^*; \hat{\beta}_{t-1}) \\
&= \frac{1}{W_t}\sum_{j=1}^{t} w_{t,j}(\nabla \ell_j(\hat{\beta}_{t-1}) - \nabla \ell_j(\beta^*)) + \frac{1}{W_t}\sum_{j=1}^{t} w_{t,j}\nabla \ell_j(\beta^*) + (\nabla \ell_0(\beta^*) - \nabla \ell_0(\hat{\beta}_{t-1})) \\
&= \frac{1}{W_t}\sum_{j=1}^{t} w_{t,j}\nabla \ell_j(\beta^*) + \left(\frac{1}{W_t}\sum_{j=1}^{t} w_{t,j}x_j x_j - \frac{1}{t_0}\sum_{i=1}^{t_0} x_i x_i\right)(\hat{\beta}_{t-1} - \beta^*) \\
&= \frac{1}{W_t}\sum_{j=1}^{t} w_{t,j}\nabla \ell_j(\beta^*) + \left(\hat{\Lambda}_t - \hat{\Lambda}_0\right)(\hat{\beta}_{t-1} - \beta^*),
\end{aligned}$$

which implies

$$\begin{aligned}
\|\nabla L_t(\beta^*; \hat{\beta}_{t-1})\|_\infty &\le \Big\| \sum_{j=1}^{t} \frac{w_{t,j}}{W_t}\nabla \ell_j(\beta^*)\Big\|_\infty + \Big\| \left(\hat{\Lambda}_t - \hat{\Lambda}_0\right)(\hat{\beta}_{t-1} - \beta^*)\Big\|_\infty \\
&\le \Big\| \sum_{j=1}^{t} \frac{w_{t,j}}{W_t}\nabla \ell_j(\beta^*)\Big\|_\infty + 2\Big(\|\hat{\Lambda}_t - \Lambda\|_\infty + \|\hat{\Lambda}_0 - \Lambda\|_\infty\Big)\|\hat{\beta}_{t-1} - \beta^*\|_1.
\end{aligned}$$

It then suffices to apply Proposition 3.7. We conclude that with probability at least $1 - 4\delta$, there exists a constant $c > 0$ such that

$$\|\nabla L_t(\beta^*; \hat{\beta}_{t-1})\|_\infty \le c z_t \sqrt{\log(p/\delta)}\sqrt{\log(pt/\delta)} + 2c\sqrt{\log(p^2/\delta)}\Big(z_t + \frac{1}{t_0^{1/2}}\Big)\|\hat{\beta}_{t-1} - \beta^*\|_1.$$

This completes the proof. □

A.4 Proof of Theorem 3.10

The following lemma can be proved by induction, and we omit its proof.

Lemma A.1. Assume that a sequence Δ_t satisfies the following recursive relationship

$$\Delta_t \le b_t + a_t \Delta_{t-1},$$

then we have

$$\Delta_t \le \sum_{j=1}^{t}\left(\prod_{k=j+1}^{t} a_k\right) b_j + \left(\prod_{j=1}^{t} a_j\right)\Delta_0.$$

Proof of Theorem 3.10. **(a)** We combine Lemma 3.4 and Proposition 3.8 to prove the desired result. Let $\Delta_j = \|\hat{\beta}_j - \beta^*\|_1$ and set

$$a_j = \frac{96sc}{\kappa}\sqrt{\log(p^2/\delta)}\left(\frac{c_0}{j^{1/2}} + \frac{1}{t_0^{1/2}}\right) \text{ and } b_j = \frac{48sc}{\kappa}\sqrt{\frac{\log(pt/\delta)}{j}}\sqrt{\log(p/\delta)}.$$

We need to upper bound

$$\sum_{j=1}^{t}\left(\prod_{k=j+1}^{t} a_k\right) b_j + \left(\prod_{j=1}^{t} a_j\right)\Delta_0.$$

First note that for $j \ge t_0$,

$$A_j = \prod_{k=j+1}^{t} a_k \le \left(\frac{96sc}{\kappa}\sqrt{\log(p^2/\delta)}\left(\frac{1 + c_0}{t_0^{1/2}}\right)\right)^{t-j}.$$

As a result,

$$\sum_{j=1}^{t} A_j b_j \le \frac{48sc}{\kappa}\sqrt{\log(pt/\delta)}\sqrt{\log(p/\delta)}\sum_{j=1}^{t}\left(\frac{96sc}{t_0^{1/2}\kappa}\sqrt{\log(p^2/\delta)}(1+c_0)\right)^{t-j}\frac{1}{\sqrt{j}}.$$

Letting $q = \frac{96sc}{t_0^{1/2}\kappa}\sqrt{\log(p^2/\delta)}(1+c_0) < 1$ and by Lemma A.2, we have

$$\sum_{j=1}^{t} q^{t-j}\frac{1}{\sqrt{j}} \le \frac{C\ln t}{\sqrt{t}}$$

for some $C > 0$. We hence obtain that with probability at least $1 - 4\delta$,

$$\|\hat{\beta}_t - \beta^*\|_1 \le \prod_{j=1}^{t} a_j \Delta_0 + \frac{48sc\ln t}{\kappa}\log(p/\delta)\sqrt{\frac{\log(pt/\delta)}{t}}.$$

Since $\prod_{j=1}^{t} a_j = \prod_{j=1}^{t_0} a_j \prod_{j=t_0+1}^{t} a_j \le \delta^{t-t_0}\prod_{j=1}^{t_0} a_j$, we can set $\prod_{j=1}^{t_0} a_j\Delta_0 = M$, which is fixed and positive. Then $\prod_{j=1}^{t} a_j\Delta_0 \le \delta^{t-t_0}M$ geometrically decays to 0. As a result, we conclude that with probability at least $1 - 4\delta$,

$$\|\hat{\beta}_t - \beta^*\|_1 \le \widetilde{\mathcal{O}}\Big(\frac{\sqrt{s}}{\kappa\sqrt{t}}\Big).$$

(b) To derive the statistical error in terms of ℓ_2-norm, we apply Lemma 3.4 and obtain

$$\|\hat{\beta}_t - \beta^*\|_2^2 \le \frac{\lambda_t + \|\nabla L_t(\beta^*;\hat{\beta}_{t-1})\|_\infty}{\kappa}\|\hat{\beta}_t - \beta^*\|_1 \le \frac{3\|\nabla L_t(\beta^*;\hat{\beta}_{t-1})\|_\infty}{\kappa}\|\hat{\beta}_t - \beta^*\|_1. \tag{9}$$

If suffices to bound the term $\|\nabla L_t(\beta^*;\hat{\beta}_{t-1})\|_\infty$. To do so, we combine Lemma 3.4 and Proposition 3.8 and obtain with probability at least $1 - 4\delta$ that for all $j \le t$

$$\begin{aligned}\|\nabla L_j(\beta^*;\hat{\beta}_{j-1})\|_\infty &\le cz_j\sqrt{\log(pt/\delta)}\sqrt{\log(p/\delta)} + 2c\sqrt{\log(p^2/\delta)}\Big(z_j + \frac{1}{t_0^{1/2}}\Big)\|\hat{\beta}_{j-1} - \beta^*\|_1 \\ &\le cz_j\sqrt{\log(pt/\delta)}\sqrt{\log(p/\delta)} + 2c\sqrt{\log(p^2/\delta)}\Big(z_j + \frac{1}{t_0^{1/2}}\Big)\frac{16s}{\kappa}\|\nabla L_{j-1}(\beta^*;\hat{\beta}_{j-2})\|_\infty.\end{aligned}$$

The condition $q = \frac{96sc}{t_0^{1/2}\kappa}\sqrt{\log(p^2/\delta)}(1+c_0) < 1$ implies $\frac{32sc}{\kappa}\sqrt{\log(p^2/\delta)}\Big(z_j + \frac{1}{t_0^{1/2}}\Big) < 1$ for $t \ge t_0$. By recursively applying the above process and adopting a similar analysis in part (a), we have

$$\|\nabla L_t(\beta^*;\hat{\beta}_{t-1})\|_\infty \le \widetilde{\mathcal{O}}\Big(\frac{1}{\kappa\sqrt{t}}\Big).$$

By combining the above result with the ℓ_1-norm error bound $\|\hat{\beta}_t - \beta^*\|_1 \le \widetilde{\mathcal{O}}(\frac{s}{\kappa\sqrt{t}})$ and (9), we conclude that with probability at least $1 - 4\delta$,

$$\|\hat{\beta}_t - \beta^*\|_2 \le \widetilde{\mathcal{O}}\Big(\frac{\sqrt{s}}{\kappa\sqrt{t}}\Big).$$

This completes the proof. □

Lemma A.2. Let $\{z_j\}$ be a sequence satisfying $z_j \le \frac{1}{\sqrt{j}}$. Then for any $q < 1$, there exists a constant $C > 0$ such that

$$\sum_{j=1}^{t} q^{t-j} z_j \le \frac{C\ln t}{\sqrt{t}}.$$

Proof of Lemma A.2. Our proof consists of two steps. In Step 1, we show that

$$\sum_{j=1}^{t} q^{t-j} z_j \le \sum_{j=1}^{t}\frac{q^{t-j}}{\sqrt{j}} \le \frac{q^{t-1}}{-2\ln q} + q^t\int_1^{t+1}\frac{1}{\sqrt{x}q^x}dx.$$

Let $f(j)=\ln(\frac{1}{q^j\sqrt{j}})$, whose derivative is $f'(j)=-\ln q-\frac{1}{2j}$. A zero of $f'(j)$ is $\hat{j}=-\frac{1}{2\ln q}$, which indicates that $f(j)$ is monotonically decreasing for $j\le\hat{j}$ and monotonically increasing for $j\ge\hat{j}$. Therefore, we have

$$\sum_{j=1}^{t}q^{t-j}z_j\le q^t\sum_{j=1}^{\lfloor\hat{j}\rfloor}\frac{1}{q^j\sqrt{j}}+q^t\int_{\lceil\hat{j}\rceil}^{t+1}\frac{1}{\sqrt{x}q^x}dx\le q^t\frac{\lfloor\hat{j}\rfloor}{q}+q^t\int_{\lceil\hat{j}\rceil}^{t+1}\frac{1}{\sqrt{x}q^x}dx\le\frac{-\frac{1}{2\ln q}}{q^{1-t}}+q^t\int_1^{t+1}\frac{1}{\sqrt{x}q^x}dx,$$

which completes Step 1.

In Step 2, we show

$$q^t\int_1^{t+1}\frac{1}{\sqrt{x}q^x}dx\le\frac{2}{q\ln(\frac{1}{q})\sqrt{t+1}}+\frac{q^{t-1}}{\ln q}+\frac{q^{\frac{3t}{4}}}{\ln(\frac{1}{q})q^{\frac{1}{4}}}.$$

We first use integration by parts and obtain

$$q^t\int_1^{t+1}\frac{1}{\sqrt{x}q^x}dx\le\frac{1}{-q\ln q\sqrt{t+1}}+\frac{q^{t-1}}{\ln q}+\frac{q^t}{2\ln(\frac{1}{q})}\int_1^{t+1}\frac{1}{\sqrt{x}^3q^x}dx,$$

where the last term in the right-hand side of the above inequality can be splitted into two parts

$$\frac{q^t}{2\ln(\frac{1}{q})}\int_1^{t+1}\frac{1}{\sqrt{x}^3q^x}dx=\frac{q^t}{2\ln(\frac{1}{q})}\int_{\frac{t+1}{4}}^{t+1}\frac{1}{\sqrt{x}^3q^x}dx+\frac{q^t}{2\ln(\frac{1}{q})}\int_1^{\frac{t+1}{4}}\frac{1}{\sqrt{x}^3q^x}dx.$$

For the first part, we have

$$\frac{q^t}{2\ln(\frac{1}{q})}\int_{\frac{t+1}{4}}^{t+1}\frac{1}{\sqrt{x}^3q^x}dx\le\frac{1}{2q\ln(\frac{1}{q})}\int_{\frac{t+1}{4}}^{t+1}\frac{1}{\sqrt{x}^3}dx=\frac{1}{q\ln(\frac{1}{q})}\frac{1}{\sqrt{t+1}}.$$

For the second part, we have

$$\frac{q^t}{2\ln(\frac{1}{q})}\int_1^{\frac{t+1}{4}}\frac{1}{\sqrt{x}^3q^x}dx\le\frac{q^{\frac{3t-1}{4}}}{2\ln(\frac{1}{q})}\int_1^{\frac{t+1}{4}}\frac{1}{\sqrt{x}^3}dx=\frac{q^{\frac{3t-1}{4}}}{\ln(\frac{1}{q})}(1-\frac{2}{\sqrt{t+1}})\le\frac{q^{\frac{3t-1}{4}}}{\ln(\frac{1}{q})}=\frac{q^{\frac{3t}{4}}}{\ln(\frac{1}{q})q^{\frac{1}{4}}}.$$

Adding the above bounds together finishes the proof in Step 2.

Combining Steps 1 and 2 acquires

$$\sum_{j=1}^{t}q^{t-j}z_j\le-\frac{q^{t-1}}{2\ln\frac{1}{q}}+\frac{2}{q\ln(\frac{1}{q})\sqrt{t+1}}+\frac{q^{\frac{3t}{4}}}{\ln(\frac{1}{q})q^{\frac{1}{4}}}\le\frac{2}{q\ln(\frac{1}{q})\sqrt{t+1}}+\frac{q^{\frac{3t}{4}}}{\ln(\frac{1}{q})q^{\frac{1}{4}}}.$$

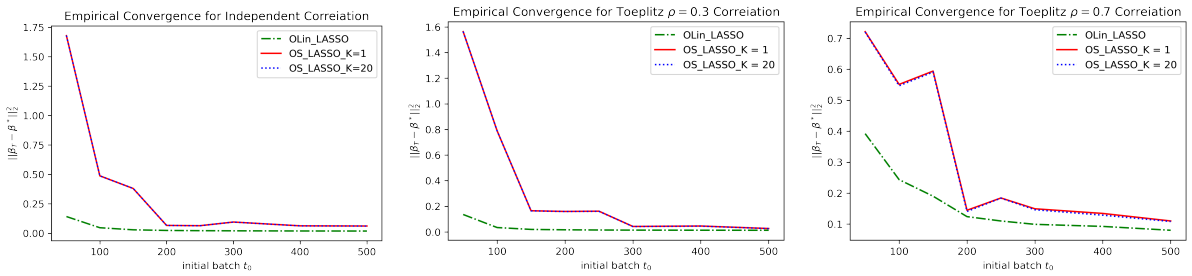
It is clear that $\frac{2}{q\ln(\frac{1}{q})\sqrt{t+1}}\le C_1\frac{\ln t}{\sqrt{t}}$ for some $C_1>0$. Note that $q^{\frac{3t}{4}}>0$, thus $g(t)=\frac{\ln(\frac{1}{q})q^{\frac{1}{4}}\ln(t)}{\sqrt{t}q^{\frac{3t}{4}}}>0$ holds for all $t\ge2$. Moreover, because $\lim_{t\to\infty}g(t)=\infty$, $g(t)$ can reach its minimum point when $t=t^*\in\{2,3,\cdots\}$. By setting $C_2=\frac{1}{g(t^*)}$, we have $C_2g(t)=C_2\frac{\ln(t)}{\sqrt{t}}/\frac{q^{\frac{3t}{4}}}{\ln(\frac{1}{q})q^{\frac{1}{4}}}\ge1$, further leading to $\frac{q^{\frac{3t}{4}}}{\ln(\frac{1}{q})q^{\frac{1}{4}}}\le C_2\frac{\ln(t)}{\sqrt{t}}$. Letting $C=C_1+C_2$, we conclude

$$\sum_{j=1}^{t}q^{t-j}z_j\le\frac{C\ln t}{\sqrt{t}},$$

which completes the proof. □

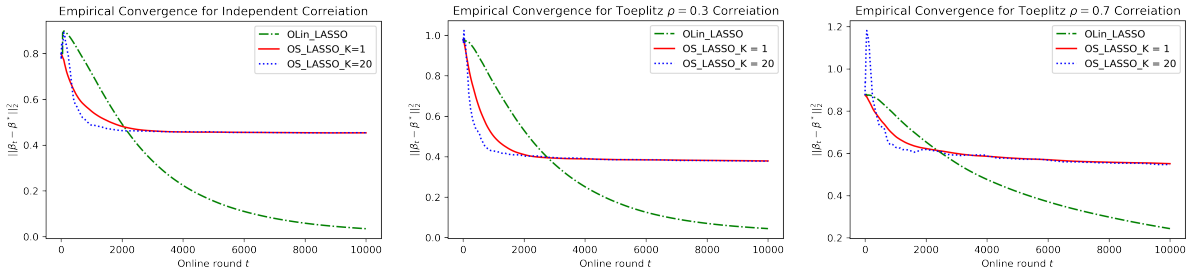
B ADDITIONAL NUMERICAL RESULTS

In this section, we conduct additional numerical experiments for instances where (i) the covariate x_i is generated under different correlation designs, (ii) the underlying β^* preserves difference sparsity levels, and (iii) the underlying β^* is generated with strong signal designs. Specifically, we consider the following three problem setups:



(a) Independent (b) Toeplitz $\rho = 0.3$ (c) Toeplitz $\rho = 0.7$

Figure 4: Empirical Convergence of MSE$\|\beta_T - \beta^*\|_2^2$ under weak signal setup for $t_0 = 50, 100, 150, \cdots, 500$ and $T = 10000$ online learning rounds for Independent, Toeplitz $\rho = 0.3$ and $\rho = 0.7$ correlation designs



(a) Independent (b) Toeplitz $\rho = 0.3$ (c) Toeplitz $\rho = 0.7$

Figure 5: Empirical Convergence of MSE$\|\beta_t - \beta^*\|_2^2$ under weak signal setup for online learning rounds $t \in [0, 10000]$ and initial batch size $t_0 = 100$ for Independent, Toeplitz $\rho = 0.3$ and $\rho = 0.7$ correlation designs

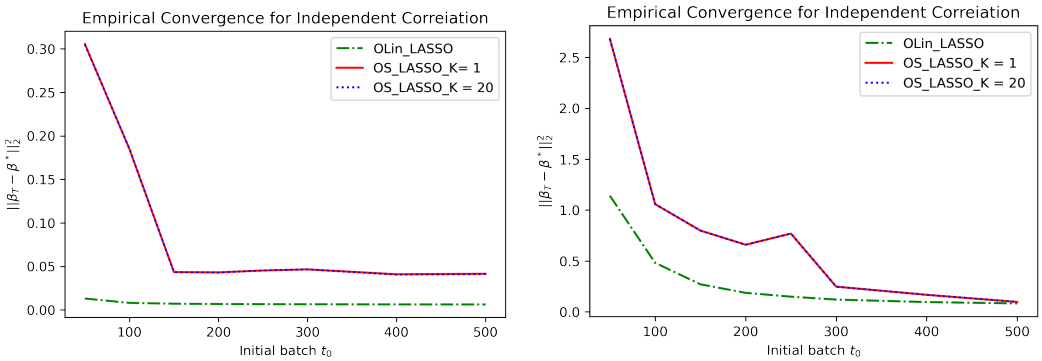
(a) The covariate x_i is generated under different correlation setups, namely independent correlation, Toeplitz $\rho = 0.3$ correlation, and $\rho = 0.7$ correlation. The underlying β^* is generated under the weak signal setup as in Section 4.

(b) The underlying β^* has different sparsity levels $s = \|\beta^*\|_0 \in \{10, 50\}$ and each nonzero entry of β^* is independently generated such that $\beta_j^* \sim \mathcal{N}(0, 0.25)$ for all $j \in \{1, 2, \cdots, s\}$. Each entry of the covariate x_i is independently generated such that $x_{ij} \sim \mathcal{N}(0, 1)$ for all $j \in [p]$.

(c) The underlying β^* has strong signals with sparsity level $s = 20$, where each nonzero entry of β^* is independently generated such that $\beta_j \sim \mathcal{N}(0, 4)$ for all $j \in S$.

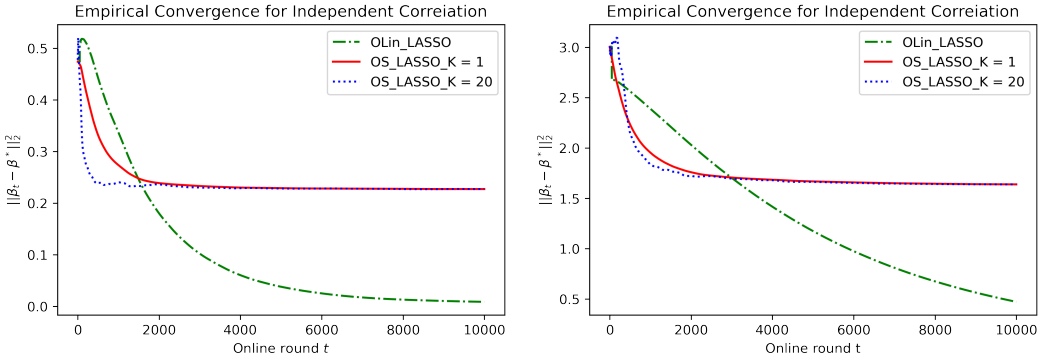
Moreover, other primitives and algorithm parameters such as step-sizes η_t are the same as those in the experiments of Section 4.

For each problem setup mentioned above, we first run our algorithm and the baselines for $T = 10^4$ online rounds with different initial batch size $t_0 = 50, 100, \cdots, 500$, and plot the empirical MSE $\|\beta_T - \beta^*\|_2^2$ against t_0. We then fix $t_0 = 100$ and plot $\|\beta_t - \beta^*\|_2^2$ against the online round $t \in [0, 10000]$. We report the results under setup (a) in Figures 4 and 5, report the results under setup (b) in Figures 6 and 7, and provide the results under setup (c) in Figure 8.



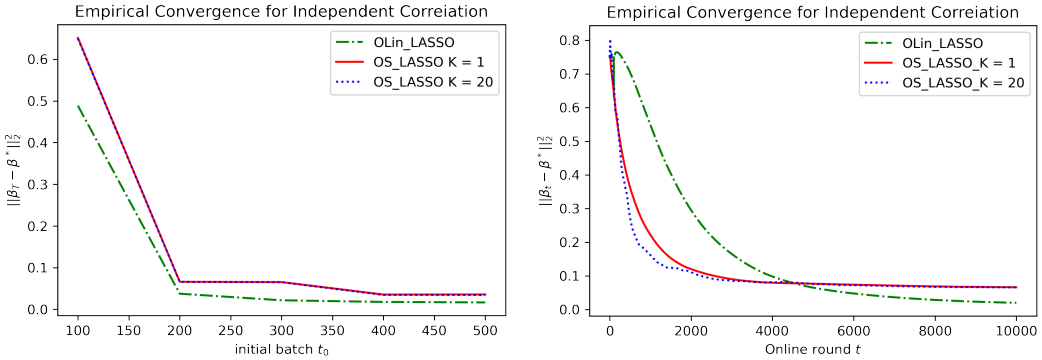
(a) sparsity: $s = 10$ (b) sparsity: $s = 50$

Figure 6: Empirical Convergence of MSE $\|\beta_T - \beta^*\|_2^2$ under weak signal setup for $t_0 = 50, 100, 150, \cdots, 500$ and $T = 10000$ online learning rounds for sparsity level $s = 10$ and $s = 50$



(a) sparsity: $s = 10$ (b) sparsity: $s = 50$

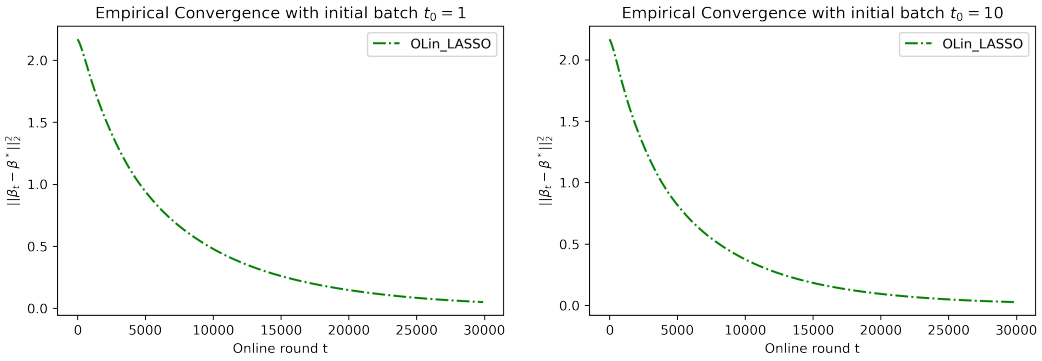
Figure 7: Empirical Convergence of MSE $\|\beta_t - \beta^*\|_2^2$ under weak signal setup for online learning rounds $t \in [0, 10000]$ and initial batch size $t_0 = 100$ for sparsity level $s = 10$ and $s = 50$



(a) online learning rounds $T = 10^4$ (b) initial batch $t_0 = 200$

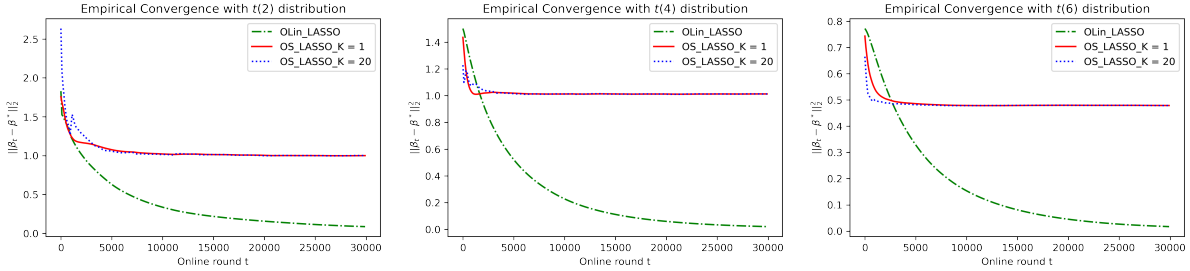
Figure 8: Empirical convergence of MSE under the strong signal setup

Figures 4 and 5 show that our algorithm outperforms the baseline algorithms under other types of covariate correlation designs. We also observe from Figures 6 and 7 that our algorithm exhibits superior performance to the baseline algorithms under different sparsity levels. These observations demonstrate the practical efficiency of our algorithm under the weak



(a) Initial batch size $t_0 = 1$ (b) Initial batch size $t_0 = 10$

Figure 9: Empirical Convergence of MSE$\|\beta_T - \beta^*\|_2^2$ under weak signal setup for $t \in [0, 3 \times 10^4]$ online learning rounds for Toeplitz $\rho = 0.3$ correlation with initial batch size $t_0 = 1$ and $t_0 = 10$ designs



(a) Noise $\epsilon \sim t_2$ (b) Noise $\epsilon \sim t_4$ (c) Noise $\epsilon \sim t_6$

Figure 10: Empirical Convergence of MSE$\|\beta_T - \beta^*\|_2^2$ under weak signal setup for $t_0 = 100$ and $t \in [0, 3 \times 10^4]$ online learning rounds for Toeplitz $\rho = 0.3$ correlation with noise $\epsilon \sim t_2$, $\epsilon \sim t_4$ and $\epsilon \sim t_6$ designs

signal setup. Further, for the strong signal setup (c), Figure 8(a) suggests that our algorithm outperforms the baselines when the initial batch size is small ($t_0 \leq 200$). However, when the initial batch size gets larger, our algorithm exhibits a comparable performance with the baselines, where the corresponding MSEs $\|\beta_T - \beta^*\|_2^2$ are around 0.03 and 0.06, respectively. This is because under strong signals, an initial batch of size $t_0 \geq 200$ is sufficient for the baseline algorithms `OS_LASSO` with different Ks to identify the true support of β^* through solving an offline LASSO using the initial batch. In contrast, as shown in Figure 2, the initial batch size $t_0 = 200$ is insufficient for the baselines to identify the true support of β^* under the weak signal setup.

To further study the performance of our algorithm with smaller initial batches, we consider the scenarios where $t_0 = 1, 10$ and report the empirical MSE in Figure 9. It can be seen that our algorithm still performs well even if the initial batch size t_0 is sufficiently small. Specially, the mean square error between the solution of our algorithm β_t and the sparse underlying coefficient β^* is 4.9×10^{-2} when $t_0 = 1$, and 2.7×10^{-2} when $t_0 = 10$ after 3×10^4 online rounds.

In addition, we consider the scenario where the noise term ϵ follows a t-distribution which has a heavy tail. We conduct three experiments where $\epsilon \sim t_2$, $\epsilon \sim t_4$ and $\epsilon \sim t_6$, respectively. For each experiment, we run our algorithm and the baselines with fixed $t_0 = 100$, and plot $\|\beta_t - \beta^*\|_2^2$ against the online round $t \in [0, 3 \times 10^4]$ in Figure 10. From Figure 10, we observe that our algorithm still generates a solution sequence converging to the sparse underlying coefficient β^* when ϵ's have heavier tails and outperforms the baseline algorithms.