

Guarantees for Multi-Step Stochastic ADMM in High Dimensions

Hanie Sedghi*

Anima Anandkumar[†]

Edmond Jonckheere[‡]

Abstract

We propose an efficient ADMM method with guarantees for high-dimensional problems. We provide explicit bounds for the sparse optimization problem and the noisy matrix decomposition problem. For sparse optimization, we establish that the modified ADMM method has an optimal regret bound of $\mathcal{O}(s \log d/T)$, where s is the sparsity level, d is the data dimension and T is the number of steps. This matches with the minimax lower bounds for sparse estimation. For matrix decomposition into sparse and low rank components, we provide the first guarantees for any online method, and prove a regret bound of $\tilde{\mathcal{O}}((s+r)\beta^2(p)/T) + \mathcal{O}(1/p)$ for a $p \times p$ matrix, where s is the sparsity level, r is the rank and $\Theta(\sqrt{p}) \leq \beta(p) \leq \Theta(p)$. Our guarantees match the minimax lower bound with respect to s, r and T . In addition, we match the minimax lower bound with respect to the matrix dimension p , i.e. $\beta(p) = \Theta(\sqrt{p})$, for many important statistical models including the independent noise model, the linear Bayesian network and the latent Gaussian graphical model under some conditions. Our ADMM method is based on epoch-based annealing and consists of inexpensive steps which involve projections on to simple norm balls.

Keywords: Stochastic ADMM, ℓ_1 regularization, multi block ADMM, sparse+low rank decomposition, regret bounds, high dimensional regime.

1 Introduction

Stochastic optimization techniques have been extensively employed for online machine learning on data which is uncertain, noisy or missing. Typically it involves performing a large number of inexpensive iterative updates, making it scalable for large-scale learning. In contrast, traditional batch-based techniques involve far more expensive operations for each update step. Stochastic optimization has been analyzed in a number of recent works, e.g. (Shalev-Shwartz, 2011; Boyd et al., 2011; Agarwal et al., 2012b; Wang et al., 2013a; Johnson and Zhang, 2013; Shalev-Shwartz and Zhang, 2013).

The alternating direction method of multipliers (ADMM) is a popular method for online and distributed optimization on a large scale (Boyd et al., 2011), and is employed in many applications, e.g. (Wahlberg et al., 2012), (Esser et al., 2010), (Mota et al., 2012). It can be viewed as a decomposition procedure where solutions to sub-problems are found locally, and coordinated via constraints to find the global solution. Specifically, it is a form of augmented Lagrangian method which applies partial updates to the dual variables. ADMM is often applied to solve regularized

*University of Southern California, Email: hsedghi@usc.edu

[†]University of California, Irvine, Email: a.anandkumar@uci.edu

[‡]University of Southern California, Email: jonckhee@usc.edu

Method	Assumptions	convergence	contraction
ST-ADMM (Ouyang et al., 2013)	L, convexity	$\mathcal{O}(d^2/\sqrt{T})$	No
ST-ADMM (Ouyang et al., 2013)	SC, in Expectation	$\mathcal{O}(d^2 \log T/T)$	No
BADMM (Wang and Banerjee, 2013)	convexity, in Expectation	$\mathcal{O}(d^2/\sqrt{T})$	No
RADAR (Agarwal et al., 2012b)	LL, LSC	$\mathcal{O}(s \log d/T)$	Yes
REASON 1 (this paper)	LSC	$\mathcal{O}(s \log d/T)$	Yes
Minimax bound (Raskutti and Yu, 2011)	Eigenvalue conditions	$\mathcal{O}(s \log d/T)$	-

Table 1: *Comparison of online sparse optimization methods under s sparsity level for the optimal paramter, d dimensional space, and T number of iterations.*

SC = Strong Convexity, LSC = Local Strong Convexity, LL = Local Lipschitz, L = Lipschitz property.

The last row provides minimax bound on error for any method. The results hold with high probability unless otherwise mentioned.

problems, where the function optimization and regularization can be carried out locally, and then coordinated globally via constraints.

Regularized optimization problems are especially relevant in the high dimensional regime since regularization is a natural mechanism to overcome ill-posedness and to encourage parsimony in the optimal solution, e.g. sparsity and low rank. Due to the efficiency of ADMM in solving regularized problems, we consider them in this paper, and provide tight guarantees for their convergence rates in the high dimensional regime.

Summary of Results and Related Work

In this paper, we consider a modified version of the stochastic ADMM method for high-dimensional problems. We first analyze the simple setting, where the optimization problem consists of a loss function and a single regularizer, and then extend to the multi-block setting with multiple regularizers and multiple variables. For illustrative purposes, for the first setting, we consider the sparse optimization problem and for the second setting, the matrix decomposition problem respectively. Note that our results easily extend to other settings, e.g. those in (Negahban et al., 2012).

For the sparse optimization problem, ℓ_1 regularization is employed and the underlying true parameter is assumed to be sparse, with sparsity level $s \ll d$, where d is the data dimension. This is a well-studied problem in a number of works (for details, refer to (Agarwal et al., 2012b)). In Table 1, we compare the convergence rates for our proposed ADMM algorithm with previous results. We prove a convergence rate of $\mathcal{O}(s \log d/T)$ in T steps, and our bound has the best of both worlds: efficient high-dimensional scaling (as $\log d$) and efficient convergence rate (as $1/T$). This also matches the minimax bound, derived by Raskutti and Yu (2011), which implies that our guarantee is unimproveable by any (batch or online) algorithm (up to constant factors). The previous stochastic ADMM methods¹ scale poorly with the data dimension (as $\mathcal{O}(d^2)$), and also have slow convergence under noisy samples (as $\mathcal{O}(1/\sqrt{T})$). Agarwal et al. (2012b) propose an

¹Goldstein et al. (2012), Deng (2012) and Luo (2012) provide faster rates for stochastic ADMM, than the rate noted in Table 1. However, they require strong conditions which are not satisfied for the optimization problems considered here, e.g. Goldstein et al. (2012) require both the loss function and the regularizer are strongly convex.

Method	Assumptions	Convergence rate
Multi-block-ADMM (Wang et al., 2013b)	L, SC, in Expectation	$\mathcal{O}(p^4 \log T/T)$
Batch method (Agarwal et al., 2012a)	LL, LSC, DF	$\mathcal{O}((s \log p + rp)/T) + \mathcal{O}(1/p^2)$
REASON 2 (this paper)	LSC, DF	$\mathcal{O}((s + r)\beta^2(p) \log p/T) + \mathcal{O}(1/p)$
Minimax bound (Agarwal et al., 2012a)	DF	$\mathcal{O}((s \log p + rp)/T) + \mathcal{O}(1/p^2)$

Table 2: *Comparison of optimization methods for sparse+low rank matrix decomposition for a $p \times p$ matrix under s sparsity level and r rank matrices and T is the number of samples.*

SC = Strong Convexity, LSC = Local Strong Convexity, LL = Local Lipschitz, L = Lipschitz for loss function, DF = diffuse low rank matrix under the optimal parameter. $\beta(p) = \Omega(\sqrt{p})$, $\mathcal{O}(p)$ and its value depends the model. The last row provides minimax bound on error for any method under the independent noise model. The results hold with high probability unless otherwise mentioned.

For Multi-block-ADMM the regret bound is on the difference of loss function from optimal loss, for the rest of works in the table, the regret bound is on $\|\tilde{S}(T) - S^\|_{\mathbb{F}}^2 + \|\tilde{L}(T) - L^*\|_{\mathbb{F}}^2$.*

efficient online method based on dual averaging, which achieves the same optimal rates as the ones derived in this paper. The main difference is that Agarwal et al. (2012b) require the function to be (locally) Lipschitz, while we do not require this condition.

We then consider the matrix decomposition problem into sparse and low rank components, and propose a modified version of the multi-block ADMM algorithm. See Table 2 for a comparison of the guarantees. To the best of our knowledge, online guarantees for high-dimensional matrix decomposition have not been provided before. Wang et al. (2013b) propose a multi-block ADMM method for the matrix decomposition problem but only provide convergence rate analysis in expectation and it has poor high dimensional scaling (as $\mathcal{O}(p^4)$ for a $p \times p$ matrix) without further modifications.

We compare our guarantees in the online setting with the batch guarantees of Agarwal et al. (2012a). Although other batch analyses exist for matrix decomposition, e.g. (Chandrasekaran et al., 2011; Candès et al., 2011; Hsu et al., 2011), they require stronger assumptions based on incoherence conditions for recovery, which we do not impose here. The batch analysis by Agarwal et al. (2012a) requires fairly mild condition such as “diffusivity” of the unknown low rank matrix. Moreover, the regret bound for the batch setting by Agarwal et al. (2012a) achieves the minimax lower bound (under the independent noise model), and is thus, optimal, up to constant factors.

As seen in Table 2, we establish efficient bounds for the modified online ADMM method. We obtain the optimal convergence rate of $\mathcal{O}(1/T)$, and a linear dependence on sparsity level s and rank r , which matches the minimax lower bound, and the batch bound of Agarwal et al. (2012a). However, our scaling with respect to the matrix dimension p depends on the problem setting. In the best case, we have $\mathcal{O}(p)$ scaling, which matches the batch bound, and in the worst case, it is $\mathcal{O}(p^2)$. The actual scaling depends on the statistical model of the noise and the loss function under consideration. We exhibit many scenarios where we match the batch bound. Under a square loss function, this includes the independent noise model (on the entries of the matrix), and a linear Bayesian network with some natural constraints, such as having a mixing

matrix with bounded spectral norm. Under the Gaussian negative log-likelihood loss function, this includes latent Gaussian graphical models with some natural constraints on the hidden and observed variables. In these scenarios, bounds from the random matrix theory (Vershynin, 2010) give us the $\mathcal{O}(p)$ scaling, allowing us to obtain the optimal convergence rate, which matches the batch bound in all the problem parameters (including p).

Note that the matrix decomposition problem suffers from an approximation error, i.e. an error even in the noiseless setting, when only the weak diffusivity condition is assumed. Both the minimax and the batch bounds in (Agarwal et al., 2012a) have an approximation error. However, our approximation error is worse by a factor of p , although it is still decaying with respect to p .

We now describe modifications to the popular stochastic ADMM method which enables us to achieve the efficient high-dimensional scaling bounds in Tables 1 and 2. This involves incorporating epoch-based annealing, on lines of (Agarwal et al., 2012b), but with a number of careful modifications. Specifically, we consider a series of time epochs with different levels of regularizations. For the sparse optimization problem, the introduced regularization constrains the optimal solution at each step to be within an ℓ_1 -norm ball of the initial estimate, at the beginning of each epoch. At the end of the epoch, an average is computed and passed on to the next epoch. The radii of the norm balls shrink over time, allowing us to move closer to the underlying true parameters, as we obtain more samples over time. We employ the inexact ADMM method, which only uses the gradient of the loss function for optimization (Wang and Banerjee, 2013). To this, we add an additional ℓ_1 -norm constraint, which decreases over epochs, and can be viewed as a form of annealing. Note that this program is a projection on to the ℓ_1 ball, and can be solved efficiently in linear time, and can also be parallelized easily (Duchi et al., 2008). For the sparse + low rank decomposition, the ADMM implementation is more involved and has multiple blocks. We apply the same principle of constraining the sparse and low rank estimates within certain norm balls, which are decreased over epochs. Even in this setting, all the updates are simple operations, consisting of closed form updates or projections on to ℓ_1 and $\|\cdot\|_*$ balls, where $\|\cdot\|_*$ denotes the nuclear norm. Note that all these operations can be performed efficiently (Duchi et al., 2008; Sra, 2010).

Our proof involves the following high-level steps to establish the regret bounds: (1) deriving regret bounds for the modified ADMM method at the end of one epoch, where the ADMM estimate is compared with the batch estimate, (2) comparing the batch estimate with the true parameter, and then combining the two steps, and analyzing over multiple epochs to obtain the final bound. For the matrix decomposition problem, additional care is needed to ensure that the errors in estimating the sparse and low rank parts can be decoupled. This is especially non-trivial in our setting since we utilize multiple variables in different blocks which are updated in each iteration. Our careful analysis enables us to establish the first results for online matrix decomposition in the high-dimensional setting which matches the batch guarantees for many interesting statistical models.

1.1 Notation

In the sequel, we use lower case letter for vectors and upper case letter for matrices. Unless otherwise stated, $x \in \mathbb{R}^d$ and $d = p^2$. Moreover, $X \in \mathbb{R}^{p \times p}$. $\|x\|_1$, $\|x\|_2$ refer to ℓ_1, ℓ_2 vector norms respectively. The term $\|X\|_*$ stands for nuclear norm of X . In addition, $\|X\|_2$, $\|X\|_F$ denote spectral and Frobenius norms respectively. We use vectorized ℓ_1, ℓ_∞ norm for matrices. i.e., $\|X\|_1 = \sum_i \sum_j |X_{ij}|$, $\|X\|_\infty = \max_{i,j} |X_{ij}|$. Finally $\rho(\Sigma) = \max_j \Sigma_{jj}$.

2 Problem Formulation

Consider the optimization problem

$$\theta^* \in \arg \min_{\theta \in \Omega} \mathbb{E}[f(\theta, x)], \quad (1)$$

where $x \in \mathbb{X}$ is a random variable and $f : \Omega \times \mathbb{X} \rightarrow \mathbb{R}$ is a given loss function. Since only samples are available, we employ the empirical estimate of $\hat{f}(\theta) := 1/n \sum_{i \in [n]} f(\theta, x_i)$ in the optimization. For high-dimensional θ , we need to impose a regularization $\mathcal{R}(\cdot)$, and

$$\hat{\theta} := \arg \min \{\hat{f}(\theta) + \lambda_n \mathcal{R}(\theta)\}, \quad (2)$$

is the batch optimal solution.

For concreteness we focus on sparse optimization and the matrix decomposition problem. It is straightforward to generalize our results to other settings, say (Negahban et al., 2012). For the first case, the optimum θ^* is a s -sparse solution, and the regularizer is the ℓ_1 norm, and we have

$$\hat{\theta} = \arg \min \{\hat{f}(\theta) + \lambda_n \|\theta\|_1\} \quad (3)$$

We also consider the matrix decomposition problem, where the underlying matrix $M^* = S^* + L^*$ is a combination of a sparse matrix S^* and a low rank matrix L^* . Here the unknown parameters are $[S^*; L^*]$, and the regularization $\mathcal{R}(\cdot)$ is a combination of the ℓ_1 norm, and the nuclear norm $\|\cdot\|_*$ on the sparse and low rank parts respectively. The corresponding batch estimate is given by

$$\begin{aligned} \{\hat{S}, \hat{L}\} &= \arg \min \{\hat{f}(M) + \lambda_n \|S\|_1 + \mu_n \|L\|_*\} \\ \text{s.t. } M &= S + L, \quad \|L\|_\infty \leq \frac{\alpha}{p}. \end{aligned} \quad (4)$$

The $\|\cdot\|_\infty$ constraint on the low rank matrix will be discussed in detail later, and it is assumed that the true matrix L^* satisfies this condition.

We consider an online version of the optimization problem where we optimize the program in (2) under each data sample instead of using the empirical estimate of f for an entire batch. We consider an inexact version of the online ADMM method, where we compute the gradient $\hat{g}_i \in \nabla f(\theta, x_i)$ at each step and employ it for optimization. In addition, we consider an epoch based setting, where we constrain the optimal solution to be close to the initial estimate at the beginning of the epoch. This can be viewed as a form of regularization and we constrain more (i.e. constrain the solution to be closer) as time goes by, since we expect to have a sharper estimate of the optimal solution. This limits the search space for the optimal solution and allows us to provide tight guarantees in the high-dimensional regime.

We first consider the simple case of sparse setting in (3), where the ADMM has double blocks, and then extend it to the sparse+low rank setting of (4), which involves multi-block ADMM.

3 ℓ_1 Regularized Stochastic Optimization

3.1 Epoch-based Online ADMM Algorithm

We now describe the modified inexact ADMM algorithm for the sparse optimization problem in (3), and refer to it as REASON 1, see Algorithm 1. We consider a sequence of epoch lengths

Algorithm 1 Regularized Epoch-based Admm for Stochastic Optimization in high-dimension 1 (REASON 1)

Input $\rho, \rho_x > 0$, epoch length schedule $\{T_i\}_{i=1}^{k_T}$, initial prox center $\tilde{\theta}_1$, initial radius R_1 , regularization parameter $\{\lambda_i\}_{i=1}^{k_T}$.

Define $\text{Shrink}_\kappa(\cdot)$ shrinkage operator in (6)

for Each epoch $i = 1, 2, \dots, k_T$ **do**

 Initialize $\theta_0 = y_0 = \tilde{\theta}_i$

for Each iteration $k = 0, 1, \dots, T_i - 1$ **do**

$$\theta_{k+1} = \arg \min_{\|\theta - \tilde{\theta}_i\|_1^2 \leq R_i^2} \{ \langle \nabla f(\theta_k), \theta - \theta_k \rangle - \langle z_k, \theta - y_k \rangle + \frac{\rho}{2} \|\theta - y_k\|_2^2 + \frac{\rho_x}{2} \|\theta - \theta_k\|_2^2 \} \quad (5a)$$

$$y_{k+1} = \text{Shrink}_{\lambda_i/\rho}(\theta_{k+1} - \frac{z_k}{\rho})$$

$$z_{k+1} = z_k - \tau(\theta_{k+1} - y_{k+1})$$

end for

Return: $\bar{\theta}(T_i) := \frac{1}{T} \sum_{k=0}^{T_i-1} \theta_k$ for epoch i and $\tilde{\theta}_{i+1} = \bar{\theta}(T_i)$.

end for

$\{T_i\}$, and in each epoch, we constrain the optimal solution to be within an ℓ_1 ball with radius R_i centered around $\tilde{\theta}_i$, which is the initial estimate of θ^* at the start of the epoch. The θ -update is given by

$$\theta_{k+1} = \arg \min_{\|\theta - \tilde{\theta}_i\|_1^2 \leq R_i^2} \{ \langle \nabla f(\theta_k), \theta - \theta_k \rangle - \langle z_k, \theta - y_k \rangle + \frac{\rho}{2} \|\theta - y_k\|_2^2 + \frac{\rho_x}{2} \|\theta - \theta_k\|_2^2 \}$$

Note that this is an inexact update since we employ the gradient $\nabla f(\cdot)$ rather than optimize directly on the loss function $f(\cdot)$ which is expensive. The above program can be solved efficiently since it is a projection on to the ℓ_1 ball, whose complexity is linear in the sparsity level of the gradient, when performed serially, and $\mathcal{O}(\log d)$ when performed in parallel using d processors (Duchi et al., 2008). For details of θ -update implementation see Appendix E.1.

For the regularizer, we introduce the variable y , and the y -update is

$$y_{k+1} = \arg \min \{ \lambda_i \|y_k\|_1 - \langle z_k, \theta_{k+1} - y \rangle + \frac{\rho}{2} \|\theta_{k+1} - y\|_2^2 \}$$

This update can be simplified to the form given in REASON 1, where $\text{Shrink}_\kappa(\cdot)$ is the soft-thresholding or shrinkage function (Boyd et al., 2011).

$$\text{Shrink}_\kappa(a) = (a - \kappa)_+ - (-a - \kappa)_+ \quad (6)$$

Thus, each step in the update is extremely simple to implement. When an epoch is complete, we carry over the average $\bar{\theta}(T_i)$ as the next epoch center and reset the other variables.

3.2 High-dimensional Guarantees

We now provide convergence guarantees for the proposed method under the following assumptions.

Assumption A1: Local strong convexity (LSC): The function $f : S \rightarrow R$ satisfies an R -local form of strong convexity (LSC) if there is a non-negative constant $\gamma = \gamma(R)$ such that

$$f(\theta_1) \geq f(\theta_2) + \langle \nabla f(\theta_2), \theta_1 - \theta_2 \rangle + \frac{\gamma}{2} \|\theta_2 - \theta_1\|_2^2.$$

for any $\theta_1, \theta_2 \in S$ with $\|\theta_1\|_1 \leq R$ and $\|\theta_2\|_1 \leq R$.

Note that the notion of strong convexity leads to faster convergence rates in general. Intuitively, strong convexity is a measure of curvature of the loss function, which relates the reduction in the loss function to closeness in the variable domain. Assuming that the function f is twice continuously differentiable, it is strongly convex, if and only if its Hessian is positive semi-definite, for all feasible θ . However, in the high-dimensional regime, where there are fewer samples than data dimension, the Hessian matrix is often singular and we do not have global strong convexity. A solution is to impose local strong convexity which allows us to provide guarantees for high dimensional problems. The notion of local strong convexity has been exploited before in a number of works on high dimensional analysis, e.g. (Negahban et al., 2012; Agarwal et al., 2012b,a).

Assumption A2: Sub-Gaussian stochastic gradients:

Let $e_k(\theta) := \nabla f(\theta, x_k) - \mathbb{E}[\nabla f(\theta, x_k)]$. For all θ such that $\|\theta - \theta^*\|_1 \leq R$, there is a constant $\sigma = \sigma(R)$ such that for all $k > 0$,

$$\mathbb{E}[\exp(\|e_k(\theta)\|_\infty^2 / \sigma^2)] \leq \exp(1)$$

Remark: The bound holds with $\sigma = \mathcal{O}(\sqrt{\log d})$ whenever each component of the error vector has sub-Gaussian tails (Agarwal et al., 2012b).

Assumption A3: Bounded dual variable: Let z denote the dual variable in ADMM. $\exists B$, s.t $\forall z, \|z\|_1 \leq B$ (Boyd et al., 2011).

The design parameters are set as

$$\begin{aligned} T_i &= C \frac{s^2}{\gamma^2} \left[\frac{\log d + B^4 + 12\sigma_i^2 \log(3/\delta_i)}{R_i^2} + 1 \right], \\ \lambda_i^2 &= \frac{\gamma}{s\sqrt{T_i}} \sqrt{R_i^2 \log d + R_i^2 B^4 + R_i^4 + 12\sigma_i^2 R_i^2 \log(3/\delta_i)} \\ \rho &= \sqrt{T_i}, \quad \rho_x = c_1 \sqrt{T_i}, \quad \tau = c_2 \sqrt{T_i}, \\ c_1 &= \frac{\sqrt{\log d}}{R_i}, \quad c_2 = 1/R_i. \end{aligned} \tag{7}$$

Theorem 1. Under assumptions A1 – A3 and parameter settings above, there exists a constant $c_0 > 0$ such that REASON 1 satisfies for all $T > k_T$,

$$\|\bar{\theta}_T - \theta^*\|_2^2 \leq c_0 \frac{s}{\gamma^2 T} [e \log d + B^4 + \sigma^2 (12 \log(6/\delta) + 24 \log k_T)] \tag{8}$$

with probability at least $1 - \delta$, where $k_T = \log_2 \frac{\gamma^2 R_1^2 T}{s^2 (\log d + B^4 + 12\sigma^2 \log(\frac{6}{\delta}))}$, and c_0 is a universal constant.

Optimal Guarantees: The above results indicate a convergence rate of $\mathcal{O}(s \log d/T)$ which matches the minimax lower bounds for sparse estimation (Raskutti and Yu, 2011). This implies that our guarantees are *unimprovable* up to constant factors.

Comparison with Agarwal et al. (2012b): The RADAR algorithm proposed by Agarwal et al. (2012b) also achieves a rate of $\mathcal{O}(s \log d/T)$ which matches with ours. The main difference is that they require the function f to be locally *Lipschitz*, while we do not have this requirement. Thus, our framework can handle a wider range of settings.

Remark on need for ℓ_1 constraint: We use ℓ_1 constraint in the θ -update step, while the usual ADMM method does not have such a constraint. The ℓ_1 constraint allows us to provide efficient high dimensional scaling (as $\mathcal{O}(\log d)$). Specifically, this is because one of the terms in our regret bound consists of $\langle e_k, \theta_k - \hat{\theta}_i \rangle$, where e_k is the error in the gradient (see C.2). We can use the inequality

$$\langle e_k, \theta_k - \hat{\theta}_i \rangle \leq \|e_k\|_\infty \|\theta_k - \hat{\theta}_i\|_1.$$

From assumption A2, we have a bound on $\|e_k\|_\infty = \mathcal{O}(\log d)$, and by imposing the ℓ_1 constraint, we also have a bound on the second term, and thus, we have an efficient regret bound. If instead ℓ_p penalty is imposed for some p , the error scales as $\|e(\theta)\|_q^2$, where ℓ_q is the dual norm of ℓ_p . For instance, if $p = 2$, we have $q = 2$, and the error can be as high as $\mathcal{O}(d/T)$ since $\|e(\theta)\|_2^2 \leq d\sigma$. Note that for the ℓ_1 norm, we have ℓ_∞ as the dual norm, and $\|e(\theta)\|_\infty \leq \sigma = \mathcal{O}(\sqrt{\log d})$ which leads to optimal regret bounds in the above theorem. Moreover, this ℓ_1 constraint can be efficiently implemented, as discussed in Section 3.1.

Application to Inverse Covariance Estimation: Consider a p -dimensional Gaussian random vector (x_1, \dots, x_p) with a sparse inverse covariance or precision matrix Θ^* . Consider the ℓ_1 -regularized maximum likelihood estimator (batch estimate),

$$\hat{\Theta} := \arg \min_{\Theta \succ 0} \{\text{Tr}(\hat{\Sigma}\Theta) - \log \det\{\Theta\} + \lambda_n \|\Theta\|_1\}, \quad (9)$$

where $\hat{\Sigma}$ is the empirical covariance matrix for the batch. This is a well-studied method for recovering the edge structure in a Gaussian graphical model, i.e. the sparsity pattern of Θ^* (Ravikumar et al., 2011). Note that the above loss function is not Lipschitz, since the gradient $\nabla f(x, \Theta) = xx^\top - \Theta^{-1}$ is not bounded in general. Therefore, the online method by Agarwal et al. (2012b) does not yield efficient bounds for this problem. On the other hand, our method REASON 1 does not have a dependence on the Lipschitz constant, and thus, we can handle this setting. We need an additional assumption that we initialize at an estimate which is away from a singular matrix, i.e. we require an initial ℓ_1 ball radius to satisfy $R_1 < \lambda_{\min}(\Theta^*)$, where $\lambda_{\min}(\cdot)$ represents the minimum eigenvalue¹. With this initialization, we have that the loss function is strongly convex for all Θ within the ball². Thus, the assumptions for Theorem 1 are satisfied in this setting, and hence, the guarantees are applicable. Finally note that for this problem, the gradient computation can be expensive since it involves computing the matrix inverse. However, efficient techniques for computing an approximate inverse can be employed, on lines of (Hsieh et al., 2011).

¹Since ℓ_1 norm bounds the spectral norm, there is no need to specify a different norm constraint.

²Let $Q = \{\theta \in \mathbb{R}^n : \alpha I_n \preceq \Theta \beta I_n\}$ then $-\log \det \Theta$ is strongly convex on Q with $\gamma = \frac{1}{\beta^2}$ (d'Aspremont et al., 2008).

3.2.1 Proof Ideas

1. In general, it is not possible to establish error contraction for stochastic ADMM at the end of each step. We establish error contracting at the end of certain time epochs, and we impose different levels of regularizations over different epochs. We perform an induction on the error, i.e. if the error at the end of k^{th} epoch is $\|\bar{\theta}(T_i) - \theta^*\|_2^2 \leq cR_i^2$, we show that in the subsequent epoch, it contracts as $\|\bar{\theta}(T_{i+1}) - \theta^*\|_2^2 \leq cR_i^2/2$ under appropriate choice of T_i , R_i and other design parameters. This is possible when we establish feasibility of the optimal solution θ^* in each epoch. Once this is established, it is straightforward to obtain the result in the theorem.
2. To show error contraction, we break down the error $\|\bar{\theta}(T_i) - \theta^*\|_2$ into two parts, viz., $\|\bar{\theta}(T_i) - \hat{\theta}(T_i)\|_2$ and $\|\hat{\theta}(T_i) - \theta^*\|_2$, where $\hat{\theta}(T_i)$ is the optimal batch estimate over the epoch T_i . The first term $\|\bar{\theta}(T_i) - \hat{\theta}(T_i)\|_2$ is obtained on the lines of analysis of stochastic ADMM, e.g. (Wang and Banerjee, 2013). Nevertheless, our analysis differs from that of (Wang and Banerjee, 2013), as theirs is not a stochastic method. i.e., the sampling error is not considered. In addition, the ℓ_1 constraint that we impose enables us to provide tight bounds for the high dimensional regime. The second term $\|\hat{\theta}(T_i) - \theta^*\|_2$ is obtained by exploiting the local strong convexity properties of the loss function, on lines of (Agarwal et al., 2012b). There are additional complications in our setting, since we have an auxiliary variable y for update of the regularization term. We relate the two variables through the dual variable, and use the fact that the dual variable is bounded.
3. In addition, we are able to prove bounds without assuming the Lipschitz property on the loss function, using the notion of *Bregman* divergence, on lines of (Wang and Banerjee, 2013). As a result, our method is applicable to a broader class of loss functions.

For Proof outline and detailed proof of Theorem 1 see Appendix A and C respectively.

4 Extension to Doubly Regularized Stochastic Optimization

We now consider the problem of matrix decomposition into a sparse matrix $S \in \mathbb{R}^{p \times p}$ and a low rank matrix $L \in \mathbb{R}^{p \times p}$ in (4) based on the loss function f on $M = S + L$. The batch program is given by

$$\begin{aligned} \hat{M} &:= \arg \min \{f(M) + \lambda_n \|S\|_1 + \mu_n \|L\|_*\} \\ \text{s.t. } & M = S + L, \quad \|L\|_\infty \leq \frac{\alpha}{p}. \end{aligned} \tag{10}$$

and we now design an online program based on multi-block ADMM algorithm, where the updates for M, S, L are carried out independently.

4.1 Epoch-based Multi-Block ADMM Algorithm

We now extend the ADMM method proposed in REASON 1 to multi-block ADMM. The details are in Algorithm 2, and we refer to it as REASON 2. Recall that the matrix decomposition setting assumes that the true matrix $M^* = S^* + L^*$ is a combination of a sparse matrix S^* and a low rank matrix L^* . In REASON 2, the updates for matrices M, S, L are done independently at each step.

For the M -update, the same linearization approach as in REASON 1 is used

$$M_{k+1} = \arg \min \{ \text{Tr}(\nabla f(M_k), M - M_k) - \text{Tr}(Z_k, M - S_k - L_k) + \frac{\rho}{2} \|M - S_k - L_k\|_{\mathbb{F}}^2 + \frac{\rho_x}{2} \|M - M_k\|_{\mathbb{F}}^2 \}$$

This is an unconstrained quadratic optimization with closed-form updates, as shown in REASON 2. The update rules for S , L are result of doing an inexact proximal update by considering them as a single block, which can then be decoupled as follows. For details, see Section 4.2.2.

$$\arg \min_{\|S - \tilde{S}_i\|_1^2 \leq \tilde{R}_i^2} \lambda_i \|S\|_1 + \frac{\rho}{2\tau_k} \|S - (S_k + \tau_k G_{M_k})\|_{\mathbb{F}}^2, \quad (11)$$

$$\arg \min_{\substack{\|L - \tilde{L}_i\|_*^2 \leq \tilde{R}_i^2 \\ \|L\|_{\infty} \leq \alpha/p}} \lambda_i \|L\|_* + \frac{\rho}{2\tau_k} \|L - (L_k + \tau_k G_{M_k})\|_{\mathbb{F}}^2, \quad (12)$$

where $G_{M_k} = M_{k+1} - S_k - L_k - \frac{1}{\rho} Z_k$.

As before, we consider epochs of varying lengths $\{T_i\}$ and constrain the estimates S and L around the epoch initializations \tilde{S}_i and \tilde{L}_i . We do not need to constrain the update of matrix M . We impose an ℓ_1 -norm constraint for the sparse estimate S . For the low rank estimate L , we impose a nuclear norm constraint around the epoch initialization \tilde{L}_i . Intuitively, the nuclear norm constraint, which is an ℓ_1 constraint on the singular values, encourages sparsity in the spectral domain leading to low rank estimates. In addition, we impose an ℓ_{∞} constraint of α/p on each entry of L , which is different from the update of S . Note that the ℓ_{∞} constraint is also imposed for the batch version of the problem (4) in (Agarwal et al., 2012a), and we assume that the true matrix L^* satisfies this constraint. For more discussions, see Section 4.2.

Note that each step of the method is easily implementable. The M -update is in closed form. The S -update involves optimization with projection on to the given ℓ_1 ball which can be performed efficiently (Duchi et al., 2008), as discussed in the previous section. For implementation details see Appendix E.2. For the L -update, we introduce an additional auxiliary variable Y and we have

$$\begin{aligned} L_{k+1} &= \min_{\|L - \tilde{L}_i\|_*^2 \leq \tilde{R}_i^2} \lambda_i \|L\|_* + \frac{\rho}{2\tau_k} \|L - (L_k + \tau_k G_{M_k})\|_{\mathbb{F}}^2 - \text{Tr}(U_k, L - Y_k) + \frac{\rho}{2} \|L - Y_k\|_{\mathbb{F}}^2 \\ Y_{k+1} &= \min_{\|Y\|_{\infty} \leq \alpha/p} \frac{\rho}{2} \|L_{k+1} - Y\|_{\mathbb{F}}^2 - \text{Tr}(U_k, L_{k+1} - Y) \\ U_{k+1} &= U_k - \tau(L_{k+1} - Y_{k+1}). \end{aligned}$$

The L -update can now be performed efficiently by computing a SVD, and then running the projection step (Duchi et al., 2008). Note that approximate SVD computation techniques can be employed for efficiency here, e.g. (Lerman et al., 2012). The Y -update is projection on to the infinity norm ball which can be found easily. Let $Y_{(j)}$ stand for j -th entry of vector(Y). The for any j -th entry of vector(Y), solution will be as follows

$$\begin{aligned} \text{If } |(L_{k+1} - U_k/\rho)_{(j)}| &\leq \frac{\alpha}{p}, \quad \text{then } Y_{(j)} = (L_{k+1} - U_k/\rho)_{(j)} \\ \text{Else } Y_{(j)} &= \text{sign} \left((L_{k+1} - U_k/\rho)_{(j)} - \frac{\alpha}{p} \right) \frac{\alpha}{p} \end{aligned}$$

As before, the epoch averages are computed and used as initializations for the next epoch.

Algorithm 2 Regularized Epoch-based Admm for Stochastic Optimization in high-dimension 2 (REASON 2)

Input $\rho, \rho_x > 0$, epoch length schedule $\{T_i\}_{i=1}^{K_T}$, regularizers $\{\lambda_i, \mu_i\}_{i=1}^{K_T}$, initial prox center \tilde{S}_1, \tilde{L}_1 , initial radii R_1, \tilde{R}_1 .

Define $Shrink_\kappa(a)$ shrinkage operator in (6), $G_{M_k} = M_{k+1} - S_k - L_k - \frac{1}{\rho}Z_k$.

for Each epoch $i = 1, 2, \dots, K_T$ **do**

 Initialize $S_0 = \tilde{S}_i, L_0 = \tilde{L}_i, M_0 = S_0 + L_0$

for Each iteration $k = 0, 1, \dots, T_i - 1$ **do**

$$\begin{aligned} M_{k+1} &= \frac{-\nabla f(M_k) + Z_k + \rho(S_k + L_k) + \rho_x M_k}{\rho + \rho_x} \\ S_{k+1} &= \min_{\|S - \tilde{S}_i\|_1^2 \leq R_i^2} \lambda_i \|S\|_1 + \frac{\rho}{2\tau_k} \|S - (S_k + \tau_k G_{M_k})\|_{\mathbb{F}}^2 \\ L_{k+1} &= \min_{\|L - \tilde{L}_i\|_*^2 \leq \tilde{R}_i^2} \lambda_i \|L\|_* + \frac{\rho}{2\tau_k} \|L - (L_k + \tau_k G_{M_k})\|_{\mathbb{F}}^2 + \frac{\rho}{2} \|L - Y_k - U_k/\rho\|_{\mathbb{F}}^2 \\ Y_{k+1} &= \min_{\|Y\|_\infty \leq \alpha/p} \frac{\rho}{2} \|L_{k+1} - Y - U_k/\rho\|_{\mathbb{F}}^2 \\ Z_{k+1} &= Z_k - \tau(M_{k+1} - (S_{k+1} + L_{k+1})) \\ U_{k+1} &= U_k - \tau(L_{k+1} - Y_{k+1}). \end{aligned}$$

end for

Set: $\tilde{S}_{i+1} = \frac{1}{T_i} \sum_{k=0}^{T_i-1} S_k$ and $\tilde{L}_{i+1} := \frac{1}{T_i} \sum_{k=0}^{T_i-1} L_k$

if $R_i^2 > 2(s + r + \frac{(s+r)^2}{p\gamma^2})\frac{\alpha^2}{p}$ **then**

 Update $R_{i+1}^2 = R_i^2/2, \tilde{R}_{i+1}^2 = \tilde{R}_i^2/2$

else

 STOP

end if

end for

4.2 High-dimensional Guarantees

We now provide guarantees that REASON 2 efficiently recovers both the sparse and low rank estimates in high dimensions efficiently. We need the following assumption, in addition to assumptions A1 and A2 from the previous section.

Assumption A4: Spectral Bound on the Gradient Error Let $E_k(M, X_k) := \nabla f(M, X_k) - \mathbb{E}[\nabla f(M, X_k)]$, $\|E_k\|_2 \leq \beta(p)\sigma$, where $\sigma := \|E_k\|_\infty$.

Recall from Assumption A2 that $\sigma = \mathcal{O}(\log p)$, under sub-Gaussianity. Here, we require spectral bounds in addition to $\|\cdot\|_\infty$ bound in A2.

Assumption A5: Bound on spikiness of low-rank matrix $\|L^*\|_\infty \leq \frac{\alpha}{p}$.

Intuitively, the ℓ_∞ constraint controls the “spikiness” of L^* . If $\alpha \approx 1$, then the entries of L are $\mathcal{O}(1/p)$, i.e. they are “diffuse” or “non-spiky”, and no entry is too large. When the low rank matrix

L^* has diffuse entries, it cannot be a sparse matrix, and thus, can be separated from the sparse S^* efficiently. In fact, the ℓ_∞ constraint is a weaker form of the *incoherence*-type assumptions needed to guarantee identifiability (Chandrasekaran et al., 2011) for sparse+low rank decomposition.

Assumption A6: Bounded dual variable Let Z, U denote the dual variables in REASON 2. $\exists B_1, B_2, s.t. \forall z, \|Z\|_1 \leq B_1, \|U\|_\infty \leq B_2$ (Boyd et al., 2011).

We choose algorithm parameters as follows.

$$T_i = C \left[\left(s + r + \frac{s+r}{\gamma} \right)^2 \left\{ \frac{\log p + B_1^4 + 12\beta^2(p)\sigma_i^2 \log(6/\delta)}{R_i^2} \right\} + \left(s + r + \frac{s+r}{\gamma} \right) \frac{\beta(p)B_1\sigma_i \sqrt{12 \log(6/\delta)}}{R_i^2} \right] \quad (13)$$

$$+ (s+r) \frac{B_2\alpha}{\sqrt{p}\tau R_i^2} + \sqrt{s+r} \frac{\sqrt{p}B_2}{\tau R_i},$$

$$\lambda_i^2 = \frac{\gamma}{s+r} \left[\sqrt{\frac{R_i^2 + \tilde{R}_i^2}{T_i}} (\sqrt{\log p} + B_1^2) + \frac{\beta(p)(R_i + \tilde{R}_i)}{\sqrt{T_i}} \sigma_i \sqrt{12 \log(3/\delta_i)} + \frac{\beta(p)B_1\sigma_i}{T_i} \sqrt{12 \log(3/\delta_i)} \right]$$

$$+ \frac{\alpha^2}{p^2}$$

$$\mu_i^2 = c_\mu \lambda_i^2$$

$$\rho = \sqrt{T_i}, \quad \rho_x = c_3 \sqrt{T_i}, \quad \tau = c_4 \sqrt{T_i},$$

$$c_3 = \frac{\sqrt{\log p}}{\sqrt{R_i^2 + \tilde{R}_i^2}}, \quad c_4 = \frac{1}{\sqrt{R_i^2 + \tilde{R}_i^2}}.$$

Theorem 2. Under assumptions A1–A6 and parameter settings as in (13), there exists a constant $c_0 > 0$ such that REASON 2 satisfies the following for all $T > k_T$,

$$\|\bar{S}(T) - S^*\|_{\mathbb{F}}^2 + \|\bar{L}(T) - L^*\|_{\mathbb{F}}^2 \leq \frac{c_0(s+r)}{T} \left[\log p + B_1^4 + \beta^2(p)\sigma^2(\log(6/\delta) + \log k_T) + \frac{\gamma B_2\alpha}{(s+r)\sqrt{p}\tau} \right] + \frac{B_2\sqrt{p}}{\tau T} + \frac{\alpha^2}{p}.$$

with probability at least $1 - \delta$ and

$$k_T \simeq -\log \frac{(s+r + (s+r)/\gamma)^2}{R_1^2 T} - \log \left[\log p + B_1^4 + 12\beta^2(p)\sigma^2(\log(6/\delta) + 2 \log k_T) + \frac{\gamma B_2\alpha}{(s+r)\sqrt{p}\tau} \right].$$

Scaling of $\beta(p)$: We have the following bounds $\Theta(\sqrt{p}) \leq \beta(p) \leq \Theta(p)$. This implies that the convergence rate is $\mathcal{O}((s+r)p \log p/T + \alpha^2/p)$, when $\beta(p) = \Theta(\sqrt{p})$ and when $\beta(p) = \Theta(p)$, it is $\mathcal{O}((s+r)p^2 \log p/T + \alpha^2/p)$. The upper bound on $\beta(p)$ arises trivially by converting the max-norm $\|E_k\|_\infty \leq \sigma$ to the bound on the spectral norm $\|E_k\|_2$. In many interesting scenarios, the lower bound on $\beta(p)$ is achieved, as outlined below in Section 4.2.1.

Comparison with the batch result: Agarwal et al. (2012a) consider the batch version of the same problem (4), and provide a regret bound of $\mathcal{O}(s \log p + rp)/T + \alpha^2/p^2$. This is also the minimax lower bound under the independent noise model. With respect to the convergence rate, we match their results with respect to the scaling of s and r , and also obtain a $1/T$ rate. We match the scaling with respect to p (up to a log factor), when $\beta(p) = \Theta(\sqrt{p})$ attains the lower bound, and we discuss a few such instances below. Otherwise, we are worse by a factor of p compared to the batch version. Intuitively, this is because we require different bounds on error terms E_k in the online and the batch settings. For online analysis, we need to bound $\sum_{k=1}^{T_i} \|E_k\|_2/T_i$ over each epoch, while for the batch analysis, we need to bound $\|\sum_{k=1}^{T_i} E_k\|_2/T_i$, which is smaller. Intuitively, the difference for the two settings can be explained as follows: for the batch setting, since we consider an empirical estimate, we operate on the averaged error, while we are manipulating each sample in the online setting and suffer from the error due to that sample. We can employ efficient concentration bounds for the batch case (Tropp, 2012), while for the online case, no such bounds exist in general. From these observations, we conjecture that our bounds in Theorem 2 are *unimprovable* in the online setting.

Approximation Error: Note that the optimal decomposition $M^* = S^* + L^*$ is not identifiable in general without the incoherence-style conditions (Chandrasekaran et al., 2011; Hsu et al., 2011). In this paper, we provide efficient guarantees without assuming such strong incoherence constraints. This implies that there is an *approximation error* which is incurred even in the noiseless setting due to model non-identifiability. Agarwal et al. (2012a) achieve an approximation error of α^2/p^2 for their batch algorithm. Our online algorithm has an approximation error of α^2/p , which is worse, but is still decaying with p . It is not clear if this bound can be improved by any other online algorithm.

4.2.1 Optimal Guarantees for Various Statistical Models

We now list some statistical models under which we achieve the batch-optimal rate for sparse+low rank decomposition.

1) Independent Noise Model: Assume we sample i.i.d. matrices $X_k = S^* + L^* + N_k$, where the noise N_k has independent bounded sub-Gaussian entries with $\max_{i,j} \text{Var}(N_k(i,j)) = \sigma^2$. We consider the square loss function, i.e. $\|X_k - S - L\|_{\mathbb{F}}^2$. In this case, $E_k = X_k - S^* - L^* = N_k$. From [Thm. 1.1](Vu, 2005), we have almost surely that $\|N_k\| = \mathcal{O}(\sigma\sqrt{p})$. We match the batch bound of Agarwal et al. (2012a) in this setting. Moreover, Agarwal et al. (2012a) provide a minimax lower bound for this model, and we match it as well. Thus, we achieve the optimal convergence rate for online matrix decomposition under the independent noise model.

2) Linear Bayesian Network: Consider a p -dimensional vector $y = Ah + n$, where $h \in \mathbb{R}^r$ with $r \leq p$, and $n \in \mathbb{R}^p$. The variables¹ h are hidden, and y are the observed variables. We assume that the vectors h and n are each zero-mean sub-Gaussian vectors with i.i.d entries, and are independent of one another. Let σ_h^2 and σ_n^2 be the variances for the entries of h and n respectively. Assume that A has full column rank. Without loss of generality, we assume that the columns of A are

¹We refer to each entry of a random vector as a variable.

normalized, since we can always rescale A and σ_h appropriately to obtain the same model. Let $\Sigma_{y,y}^*$ be the true covariance matrix of y . From the independence assumptions, we have $\Sigma_{y,y}^* = S^* + L^*$, where $S^* = \sigma_n^2 I$ and $L^* = \sigma_h^2 A A^\top$.

In each step k , we obtain a sample y_k from the Bayesian network. For the square loss function f , we have the error $E_k = y_k y_k^\top - \Sigma_{y,y}^*$. Applying [Cor. 5.50](Vershynin, 2010), we have, with probability $1 - e^{-cp}$, for some constant $c > 0$,

$$\|n_k n_k^\top - \sigma_n^2 I\|_2 = \mathcal{O}(\sqrt{p} \sigma_n^2), \quad \|h_k h_k^\top - \sigma_h^2 I\|_2 = \mathcal{O}(\sqrt{p} \sigma_h^2). \quad (14)$$

We thus have with probability $1 - T e^{-cp}$,

$$\|E_k\|_2 \leq \mathcal{O}(\sqrt{p}(\|A\|^2 \sigma_h^2 + \sigma_n^2)), \quad \forall k \leq T.$$

When $\|A\|_2$ is bounded, we obtain the optimal bound in Theorem 2, which matches the batch bound.

If the entries of A are *generically* drawn (e.g. from a Gaussian distribution), we have $\|A\|_2 = \mathcal{O}(1 + \sqrt{r/p})$. Moreover, such generic matrices A are also “diffuse”, and thus, the low rank matrix L^* satisfies Assumption A5, with $\alpha \sim 1$. Intuitively, when A is generically drawn, there are diffuse connections from hidden to observed variables, and we have efficient guarantees under this setting.

Thus, our online method matches the batch guarantees for linear Bayesian networks when the entries of the observed vector y are conditionally independent given the latent variables h . When this assumption is violated, the above framework is no longer applicable since the true covariance matrix $\Sigma_{y,y}^*$ is *not* composed of a sparse matrix. To handle such models, we consider matrix decomposition of the inverse covariance or the precision matrix $M^* := \Sigma_{y,y}^{*-1}$, which can be expressed as a combination of sparse and low rank matrices, for the class of latent Gaussian graphical models, described below.

3) Latent Gaussian Graphical Model Estimation: Consider the Bayesian network on p -dimensional observed variables as

$$y = A h + B y + n, \quad y, n \in \mathbb{R}^p, h \in \mathbb{R}^r, \quad (15)$$

as in Figure 1 where h, y and n are drawn from a zero-mean multivariate Gaussian distribution. The vectors h and n are independent of one another, and $n \sim \mathcal{N}(0, \sigma_n^2 I)$. Assume that A has full column rank. Without loss of generality, we assume that A has normalized columns, and that h has independent entries (Pitman and Ross, 2012). For simplicity, let $h \sim \mathcal{N}(0, \sigma_h^2 I)$ (more generally, its covariance is a diagonal matrix). Note that the matrix $B = 0$ in the previous setting (the previous setting allows for more general sub-Gaussian distributions, and here, we limit ourselves to the Gaussian distribution). For the model in (15), the precision matrix M^* with respect to the marginal distribution on the observed vector y is given by

$$M^* = \Sigma_{y,y}^{*-1} = \widetilde{M}_{y,y}^* - \widetilde{M}_{y,h}^* (\widetilde{M}_{h,h}^*)^{-1} \widetilde{M}_{h,y}^*, \quad (16)$$

where $\widetilde{M}^* = \Sigma^{*-1}$, and Σ^* is the joint-covariance matrix of vectors y and h . It is easy to see that the second term in (16) has rank at most r . The first term in (16) is sparse under some natural constraints, viz., when the matrix B is sparse, and there are a small number of *colliders* among the observed variables y . A triplet of variables consisting of two *parents* and their *child* in

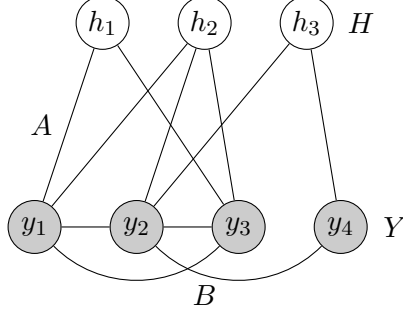


Figure 1: Graphical representation of a latent variable model.

a Bayesian network is termed as a collider. The presence of colliders results in additional edges when the Bayesian network on y and h is converted to an undirected graphical model, whose edges are given by the sparsity pattern $\widetilde{M}_{y,y}^*$, the first term in (16). Such a process is known as *moralization* (Lauritzen, 1996), and it involves introducing new edges between the parents in the directed graph (the graph of the Bayesian networks), and removing the directions to obtain an undirected model. Thus, when the matrix B is sparse, and there are a small number of colliders among the observed variables y , the resulting sub-matrix $\widetilde{M}_{y,y}^*$ is also sparse.

We thus have the precision matrix M^* in (16) as $M^* = S^* + L^*$, where S^* and L^* are sparse and low rank components. We can find this decomposition via regularized maximum likelihood. The batch estimate is given by (Chandrasekaran et al., 2012)

$$\{\hat{S}, \hat{L}\} := \arg \min \{ \text{Tr}(\hat{\Sigma}_n M) - \log \det M + \lambda_n \|S\|_1 + \mu_n \|L\|_* \}, \quad (17)$$

$$s.t. \quad M = S + L. \quad (18)$$

This is a special case of (4) with the loss function $f(M) = \text{Tr}(\hat{\Sigma}_n M) - \log \det M$. In this case, we have the error $E_k = y_k y_k^\top - M^{*-1}$. Since $y = (I - B)^{-1}(Ah + n)$, we have the following bound w.h.p.

$$\|E_k\|_2 \leq \mathcal{O} \left(\frac{\sqrt{p} \cdot (\|A\|_2^2 \sigma_h^2 + \sigma_n^2) \log(pT)}{\sigma_{\min}(I - B)^2} \right), \quad \forall k \leq T,$$

where $\sigma_{\min}(\cdot)$ denotes the minimum singular value. The above result is obtained by alluding to (14).

When $\|A\|_2$ and $\sigma_{\min}(I - B)$ are bounded, we thus achieve optimal scaling for our proposed online method. As discussed for the previous case, when A is generically drawn, $\|A\|_2$ is bounded. To bound $\sigma_{\min}(I - B)$, a sufficient condition is *walk-summability* on the sub-graph among the observed variables y . The class of walk-summable models is efficient for inference (Malioutov et al., 2006) and structure learning (Anandkumar et al., 2012), and they contain the class of attractive models. Thus, it is perhaps not surprising that we obtain efficient guarantees for such models for our online algorithm.

We need to slightly change the algorithm REASON 2 for this scenario as follows: for the M -update in REASON 2, we add a Frobenius norm constraint on M as $\|M_k - \tilde{S}_i - \tilde{L}_i\|_{\mathbb{F}}^2 \leq \check{R}^2$, and this can still be computed efficiently, since it involves projection on to the ℓ_2 norm all, see Appendix E.1. We assume a good initialization M which satisfies $\|M - M^*\|_{\mathbb{F}}^2 \leq \check{R}^2$. This ensures that M_k in subsequent steps is non-singular, and that the gradient of the loss function f in (17), which

involves M_k^{-1} , can be computed. As observed in section 3.2 on sparse graphical model selection, the method can be made more efficient by computing approximate matrix inverses Hsieh et al. (2013). As observed before, the loss function f satisfies the local strong convexity property, and the guarantees in Theorem 2 are applicable. Note that the batch analysis of Agarwal et al. (2012a) cannot handle this graphical model scenario, since the loss function in (17) is not (locally) Lipschitz, as required by Agarwal et al. (2012a).

4.2.2 Proof Ideas

We now provide a short overview of proof techniques for establishing the guarantees in Theorem 2. It builds on the proof techniques used for proving Theorem 1, but is significantly more involved since we now need to decouple the errors for sparse and low rank matrix estimation, and our ADMM method consists of multiple blocks. The main steps are as follows

1. It is convenient to define $W = [S; L]$ to merge the variables L and S into a single variable W , as in (Ma et al., 2012). Let $\phi(W) = \|S\|_1 + \frac{\mu_i}{\lambda_i} \|L\|_*$, and $A = [I, I]$. The ADMM update for S and L in REASON 2, can now be rewritten as a single update for variable W . Consider the update

$$W_{k+1} = \arg \min_W \{ \lambda_i \phi(W) + \frac{\rho}{2} \|M_{k+1} - AW - \frac{1}{\rho} Z_k\|_{\mathbb{F}}^2 \}$$

The above problem is not easy to solve as the S and L parts are coupled together. Instead, we solve it inexactly through one step of a proximal gradient method as in (Ma et al., 2012) as

$$\arg \min_W \{ \lambda_i \phi(W) + \frac{\rho}{2\tau_k} \|W - [W_k + \tau_k A^\top (M_{k+1} - AW_k - \frac{1}{\rho} Z_k)]\|_{\mathbb{F}}^2 \}. \quad (19)$$

Since the two parts of $W = [S; L]$ are separable in the quadratic part now, Equation (19) reduces to two decoupled updates on S and L as given by (11) and (12).

2. It is convenient to analyze the W update in Equation (19) to derive regret bounds for the online update in one time epoch. Once this is obtained, we also need error bounds for the batch procedure, and we employ the guarantees from Agarwal et al. (2012a). As in the previous setting of sparse optimization, we combine the two results to obtain an error bound for the online updates by considering multiple time epochs.
3. An added difficulty in the matrix decomposition problem is decoupling the errors for the sparse and low rank estimates. To this end, we impose norm constraints on the estimates of S and L , and carry them over from epoch to epoch. On the other hand, at the end of each epoch M is reset. Special care needs to be taken in many steps of the proof to carefully transform the various norm bounds, where a naive analysis would lead to worse scaling in the dimensionality p . We instead carefully project the error matrices on to on and off support of S^* for the ℓ_1 norm term, and similarly onto the range and its complement of L^* for the nuclear norm term. This allows us to have a convergence rate with a $s + r$ term, in place of p .

Thus, our careful analysis leads to tight guarantees for online matrix decomposition. For Proof outline and detailed proof of Theorem 2 see Appendix B and D respectively.

Acknowledgment

The authors thank Alekh Agarwal for detailed discussions about his work and the minimax bounds. The authors thank Majid Janzamin for valuable comments on sparse and sparse + low rank recovery. A. Anandkumar is supported in part by Microsoft Faculty Fellowship, NSF Career award CCF-1254106, NSF Award CCF-1219234, and ARO YIP Award W911NF-13-1-0084.

References

- A. Agarwal, S. Negahban, and M. Wainwright. Noisy matrix decomposition via convex relaxation: Optimal rates in high dimensions. *The Annals of Statistics*, 40(2):1171–1197, 2012a.
- A. Agarwal, S. Negahban, and M. J. Wainwright. Stochastic optimization and sparse statistical recovery: Optimal algorithms for high dimensions. In *NIPS*, pages 1547–1555, 2012b.
- A. Anandkumar, V. Tan, F. Huang, and A.S. Willsky. High-dimensional gaussian graphical model selection: walk summability and local separation criterion. *Journal of Machine Learning*, 13: 22932337, August 2012.
- S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122, 2011.
- E. J. Candès, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *Journal of the ACM (JACM)*, 58(3):11, 2011.
- V. Chandrasekaran, S. Sanghavi, Pablo A Parrilo, and A. S Willsky. Rank-sparsity incoherence for matrix decomposition. *SIAM Journal on Optimization*, 21(2):572–596, 2011.
- V. Chandrasekaran, P. A Parrilo, and Alan S Willsky. Latent variable graphical model selection via convex optimization. *The Annals of Statistics*, 40(4):1935–1967, 2012.
- Alexandre d’Aspremont, Onureena Banerjee, and Laurent El Ghaoui. First-order methods for sparse covariance selection. *SIAM Journal on Matrix Analysis and Applications*, 30(1):56–66, 2008.
- W. Deng, W. and Yin. On the global and linear convergence of the generalized alternating direction method of multipliers. Technical report, DTIC Document, 2012.
- J. Duchi, S. Shalev-Shwartz, Y. Singer, and T. Chandra. Efficient projections onto the l_1 -ball for learning in high dimensions. In *Proceedings of the 25th international conference on Machine learning*, pages 272–279. ACM, 2008.
- E. Esser, X. Zhang, and T. Chan. A general framework for a class of first order primal-dual algorithms for convex optimization in imaging science. *SIAM Journal on Imaging Sciences*, 3(4):1015–1046, 2010.
- T Goldstein, B. ODonoghue, and S. Setzer. Fast alternating direction optimization methods. *CAM report*, pages 12–35, 2012.

- C. Hsieh, M. A Sustik, I. Dhillon, P. Ravikumar, and R. Poldrack. Big\ & quic: Sparse inverse covariance estimation for a million variables. In *Advances in Neural Information Processing Systems*, pages 3165–3173, 2013.
- Cho-Jui Hsieh, Matyas A Sustik, Inderjit S Dhillon, and Pradeep D Ravikumar. Sparse inverse covariance matrix estimation using quadratic approximation. In *NIPS*, pages 2330–2338, 2011.
- Daniel Hsu, Sham M Kakade, and Tong Zhang. Robust matrix decomposition with sparse corruptions. *Information Theory, IEEE Transactions on*, 57(11):7221–7234, 2011.
- R. Johnson and T. Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In C.J.C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 315–323. 2013.
- S.L. Lauritzen. *Graphical models*. Clarendon Press, 1996.
- Gilad Lerman, Michael McCoy, Joel A Tropp, and Teng Zhang. Robust computation of linear models, or how to find a needle in a haystack. *arXiv preprint arXiv:1202.4044*, 2012.
- Zhi-Quan Luo. On the linear convergence of the alternating direction method of multipliers. *arXiv preprint arXiv:1208.3922*, 2012.
- S. Ma, L. Xue, and H. Zou. Alternating direction methods for latent variable gaussian graphical model selection. *arXiv preprint arXiv:1206.1275v2*, 2012.
- Dmitry M Malioutov, Jason K Johnson, and Alan S Willsky. Walk-sums and belief propagation in gaussian graphical models. *The Journal of Machine Learning Research*, 7:2031–2064, 2006.
- J. FC Mota, J. MF Xavier, P. MQ Aguiar, and M. Puschel. Distributed admm for model predictive control and congestion control. In *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*, pages 5110–5115. IEEE, 2012.
- S. Negahban, P. Ravikumar, M. Wainwright, and B. Yu. A unified framework for high-dimensional analysis of m -estimators with decomposable regularizers. *Statistical Science*, 27(4):538–557, 2012.
- H. Ouyang, N. He, L. Tran, and A. G Gray. Stochastic alternating direction method of multipliers. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pages 80–88, 2013.
- Jim Pitman and Nathan Ross. Archimedes, gauss, and stein. *Notices AMS*, 59:1416–1421, 2012.
- M. Raskutti, G. and Wainwright and B. Yu. Minimax rates of estimation for high-dimensional linear regression over j formula formulatype=. *Information Theory, IEEE Transactions on*, 57(10):6976–6994, 2011.
- P. Ravikumar, M.J. Wainwright, G. Raskutti, and B. Yu. High-dimensional covariance estimation by minimizing ℓ_1 -penalized log-determinant divergence. *Electronic Journal of Statistics*, (4): 935–980, 2011.
- S. Shalev-Shwartz. Online learning and online convex optimization. *Foundations and Trends in Machine Learning*, 4(2):107–194, 2011.

- S. Shalev-Shwartz and T. Zhang. Accelerated mini-batch stochastic dual coordinate ascent. In C.J.C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 378–385. 2013.
- Suvrit Sra. Generalized proximity and projection with norms and mixed-norms. *Technique Report, MPI*, 2010.
- J. Tropp. User-friendly tail bounds for sums of random matrices. *Foundations of Computational Mathematics*, 12(4):389–434, 2012.
- Roman Vershynin. Introduction to the non-asymptotic analysis of random matrices. *arXiv preprint arXiv:1011.3027*, 2010.
- Van H Vu. Spectral norm of random matrices. In *Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, pages 423–430. ACM, 2005.
- B. Wahlberg, S. Boyd, M. Annergren, and Y. Wang. An admm algorithm for a class of total variation regularized estimation problems. *arXiv preprint arXiv:1203.1828*, 2012.
- C. Wang, X. Chen, A. Smola, and E. Xing. Variance reduction for stochastic gradient optimization. In C.J.C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 181–189. 2013a.
- H. Wang and A. Banerjee. Bregman alternating direction method of multipliers. *arXiv preprint arXiv:1306.3203*, 2013.
- X. Wang, M. Hong, S. Ma, and Z. Luo. Solving multiple-block separable convex minimization problems using two-block alternating direction method of multipliers. *arXiv preprint arXiv:1308.5294*, 2013b.

A Proof outline for Theorem 1

The foundation block for this proof is Proposition 1.

Proposition 1. *Suppose f satisfies Assumptions A1, A2 with parameters γ and σ_i respectively and assume that $\|\theta^* - \hat{\theta}_i\|_1^2 \leq R_i^2$. We apply the updates in REASON 1 with parameters as in (7). Then, there exists a universal constant c such that for any radius R_i*

$$f(\bar{\theta}(T_i)) - f(\hat{\theta}_i) + \lambda_i \|\bar{y}(T_i)\|_1 - \lambda_i \|\hat{\theta}_i\|_1 \leq \frac{R_i \sqrt{\log d}}{\sqrt{T_i}} + \frac{R_i B^2}{\sqrt{T_i}} + \frac{R_i^2}{\sqrt{T_i}} + \frac{R_i \sigma_i \sqrt{12 \log(3/\delta_i)}}{\sqrt{T_i}}, \quad (20a)$$

$$\|\bar{\theta}(T_i) - \theta^*\|_1^2 \leq \frac{c'}{\sqrt{C}} R_i^2. \quad (20b)$$

where both bounds are valid with probability at least $1 - \delta_i$.

In order to prove Proposition 1, we need to prove some more lemmas.

To move forward from here please note the following notations: $\Delta_i = \hat{\theta}_i - \theta^*$ and $\hat{\Delta}(T_i) = \bar{\theta}_i - \hat{\theta}_i$.

Lemma 1. *At epoch i assume that $\|\theta^* - \tilde{\theta}_i\|_1 \leq R_i$. Then the error Δ_i satisfies the bounds*

$$\|\hat{\theta}_i - \theta^*\|_2 \leq \frac{4}{\gamma} \sqrt{s} \lambda_i, \quad (21a)$$

$$\|\hat{\theta}_i - \theta^*\|_1 \leq \frac{8}{\gamma} s \lambda_i. \quad (21b)$$

Lemma 2. *Under the conditions of Proposition 1 and with parameter settings (7), we have*

$$\|\hat{\Delta}(T_i)\|_2^2 \leq \frac{c'}{\sqrt{C}} \frac{1}{s} R_i^2,$$

with probability at least $1 - \delta_i$.

B Proof outline for Theorem 2

The foundation block for this proof is Proposition 2.

Proposition 2. *Suppose f satisfies Assumptions A1 – A6 with parameters γ and σ_i respectively and assume that $\|S^* - \tilde{S}_i\|_1^2 \leq R_i^2$, $\|L^* - \tilde{L}_i\|_1^2 \leq \tilde{R}_i^2$. We apply the updates in REASON 2 with parameters as in (13). Then, there exists a universal constant c such that for any radius R_i, \tilde{R}_i , $\tilde{R}_i = c_r R_i, 0 \leq c_r \leq 1$,*

$$f(\bar{M}(T_i)) + \lambda_i \phi(\bar{W}(T_i)) - f(\hat{M}_i) - \lambda_i \phi(\hat{W}(T_i)) \leq \frac{\sqrt{R_i^2 + \tilde{R}_i^2}}{T_i} \sqrt{\log p} + \frac{B_1^2 \sqrt{R_i^2 + \tilde{R}_i^2}}{\sqrt{T_i}} + \frac{\beta(p)(R_i + \tilde{R}_i + B_1) \sigma_i \sqrt{12 \log(3/\delta_i)}}{\sqrt{T_i}}, \quad (22a)$$

$$\|\bar{S}(T_i) - S^*\|_1^2 \leq \frac{c'}{\sqrt{C}} R_i^2 + c(s + r + \frac{(s + r)^2}{p \gamma^2}) \frac{\alpha^2}{p}, \quad (22b)$$

$$\|\bar{L}(T_i) - L^*\|_*^2 \leq \frac{c'}{\sqrt{C}} \frac{1}{1 + \gamma} R_i^2 + c \frac{(s + r)^2 \alpha^2}{p \gamma^2} \frac{1}{p}.$$

where both bounds are valid with probability at least $1 - \delta_i$.

In order to prove Proposition 2, we need two more lemmas.

To move forward, we use the following notations: $\Delta(T_i) = \hat{S}_i - S^* + \hat{L}_i - L^*$, $\Delta^*(T_i) = \bar{S}(T_i) - S^* + \bar{L}(T_i) - L^*$ and $\hat{\Delta}(T_i) = \bar{S}_i - \hat{S}_i + \bar{L}_i - \hat{L}_i$. In addition $\Delta_S(T_i) = \hat{S}_i - S^*$, with alike notations for $\Delta_L(T_i)$. For on and off support part of $\Delta(T_i)$, we use $(\Delta(T_i))_{supp}$ and $(\Delta(T_i))_{supp^c}$.

Lemma 3. *At epoch i assume that $\|S^* - \tilde{S}\|_1^2 \leq R_i^2$, $\|L^* - \tilde{L}\|_1^2 \leq \tilde{R}_i^2$. Then the errors $\Delta_S(T_i), \Delta_L(T_i)$ satisfy the bound*

$$\|\hat{S}_i - S^*\|_{\mathbb{F}}^2 + \|\hat{L}_i - L^*\|_{\mathbb{F}}^2 \leq c\{s\frac{\lambda_i^2}{\gamma^2} + r\frac{\mu_i^2}{\gamma^2}\}.$$

Lemma 4. *Under the conditions of Proposition 2 and with parameter settings (13), (13), we have*

$$\begin{aligned} & \|\hat{S}_i - \bar{S}(T_i)\|_{\mathbb{F}}^2 + \|\hat{L}_i - \bar{L}(T_i)\|_{\mathbb{F}}^2 \\ & \leq \frac{2}{\gamma} \left[\sqrt{\frac{R_i^2 + \tilde{R}_i^2}{T_i}} \sqrt{\log p} + \sqrt{\frac{R_i^2 + \tilde{R}_i^2}{T_i}} B_1^2 + \frac{\beta(p)(R_i + \tilde{R}_i)\sigma_i}{\sqrt{T_i}} \sqrt{12 \log(3/\delta_i)} + \frac{\beta(p)B_1\sigma_i}{T_i} \sqrt{12 \log(3/\delta_i)} \right] \\ & + \left(\frac{\alpha}{\sqrt{p}} + \frac{B_2\sqrt{p}}{\tau T_i} \right)^2. \end{aligned}$$

with probability at least $1 - \delta_i$.

C Proof of Theorem 1

The first step is to ensure that $\|\theta^* - \tilde{\theta}_i\| \leq R_i$ holds at each epoch so that Proposition 1 can be applied in a recursive manner. We prove this by induction on the epoch index. By construction, this bound holds at the first epoch. Assume that it holds for epoch i . Recall that T_i is defined by (7) where $C \geq 1$ is a constant we can choose. By substituting this T_i in inequality (20b), the simplified bound (20b) further yields

$$\|\bar{\theta}(T_i) - \theta^*\|_1^2 \leq \frac{c'}{\sqrt{C}} R_i^2.$$

Thus, by choosing C sufficiently large, we can ensure that $\|\bar{\theta}(T_i) - \theta^*\|_1^2 \leq R_i^2/2 := R_{i+1}^2$. Consequently, if θ^* is feasible at epoch i , it stays feasible at epoch $i+1$. Hence, by induction we are guaranteed the feasibility of θ^* throughout the run of algorithm.

As a result, Lemma 2 applies and we find that

$$\|\hat{\Delta}(T_i)\|_2^2 \leq \frac{c}{s} R_i^2. \quad (23)$$

We have now bounded $\hat{\Delta}(T_i) = \bar{\theta}(T_i) - \hat{\theta}_i$ and Lemma 1 provides a bound on $\Delta_i = \hat{\theta}_i - \theta^*$, such that the error $\Delta^*(T_i) = \bar{\theta}(T_i) - \theta^*$ can be controlled by triangle inequality. In particular, by combining (21a) with (23), we get

$$\|\Delta^*(T_i)\|_2^2 \leq c\left\{\frac{1}{s} R_i^2 + \frac{16}{s} R_i^2\right\},$$

i.e.

$$\|\Delta^*(T_i)\|_2^2 \leq c \frac{R_1^2 2^{-(i-1)}}{s}. \quad (24)$$

The bound holds with probability at least $1 - 3 \exp(-w_i^2/12)$. Recall that $R_i^2 = R_1^2 2^{-(i-1)}$. Since $w_i = w^2 + 24 \log i$, we can apply union bound to simplify the error probability as $1 - 6 \exp(-w^2/12)$. Throughout this paper we use $\delta_i = 3 \exp(-w_i^2/12)$ and $\delta = 6 \exp(-w^2/12)$ to simplify the equations. To complete the proof we need to convert the error bound (24) from its dependence on the number of epochs k_T to the number of iterations needed to complete k_T epochs, i.e. $T(K) = \sum_{i=1}^k T_i$.

$$\begin{aligned} T(k) &= \sum_{i=1}^k T_i = C \sum_{i=1}^k \left[\frac{s^2}{\gamma^2 R_i^2} \{\log d + B^4 + \frac{w_i^2 \sigma_i^2}{R_i^2}\} + \frac{s^2}{\gamma^2} \right] \\ &= C \sum_{i=1}^k \left[\frac{s^2}{\gamma^2 R_1^2} \{\log d + B^4 + \sigma^2(w^2 + 24 \log k)\} 2^{i-1} + \frac{s^2}{\gamma^2} \right]. \end{aligned}$$

$$T(k) \leq C \left[\frac{s^2}{\gamma^2 R_1^2} \{\log d + B^4 + \sigma^2(w^2 + 24 \log k)\} 2^k + \frac{s^2}{\gamma^2} \right].$$

$T(k) \leq S(k)$, therefore $k_T \geq S^{-1}(T)$.

$$S(k) = C \left[\frac{s^2}{\gamma^2 R_1^2} \{\log d + B^4 + \sigma^2(w^2 + 24 \log k)\} 2^k + \frac{s^2}{\gamma^2} \right].$$

Using a first order approximation for $\log(a + b)$,

$$\begin{aligned} \log(T) &\simeq \log C + k_T + \log \left[\frac{s^2}{\gamma^2 R_1^2} \{\log d + B^4 + \sigma^2(w^2 + 24 \log k)\} \right], \\ k_T &\simeq \log T - \log C - \log \left[\frac{s^2}{\gamma^2 R_1^2} \{\log d + B^4 + \sigma^2(w^2 + 24 \log k)\} \right]. \end{aligned}$$

Therefore,

$$2^{-k_T} = \frac{C s^2}{\gamma^2 T R_1^2} \{\log d + B^4 + \sigma^2(w^2 + 24 \log k)\}.$$

Putting this back into (24), we get that

$$\begin{aligned} \|\Delta^*(T_i)\|_2^2 &\leq c \frac{R_1^2}{s} \frac{C s^2}{\gamma^2 T R_1^2} \{\log d + B^4 + \sigma^2(w^2 + 24 \log k)\} \\ &\leq c \frac{s}{\gamma^2 T} \{\log d + B^4 + \sigma^2(w^2 + 24 \log k)\}. \end{aligned}$$

Using the definition $\delta = 6 \exp(-w^2/12)$, above bound holds with probability $1 - \delta$. Simplifying the error in terms of δ by replacing w^2 with $12 \log(6/\delta)$, gives us (8).

C.1 Proofs for Convergence within a Single Epoch for Algorithm 1

Lemma 5. For $\bar{\theta}(T_i)$ defined in Algorithm 1 and $\hat{\theta}_i$ the optimal value for epoch i , let $\rho = \sqrt{T_i}$, $\rho_x = c_1\sqrt{T_i}$ and $\tau = c_2\sqrt{T_i}$ where $\log d = e \log d$, $c_1 = \frac{\sqrt{\log d}}{R_i}$, $c_2 = 1/R_i$. We have that

$$f(\bar{\theta}(T_i)) - f(\hat{\theta}_i) + \lambda_i \|\bar{y}(T_i)\|_1 - \lambda_i \|\hat{\theta}_i\|_1 \leq \frac{R_i \sqrt{\log d}}{\sqrt{T_i}} + \frac{R_i B^2}{\sqrt{T_i}} + \frac{R_i^2}{\sqrt{T_i}} + \frac{\sum_{k=1}^{T_i} \langle e_k, \hat{\theta}_i - \theta_k \rangle}{T_i}. \quad (25)$$

Proof. First we show that our update rule for θ is equivalent to not linearizing f and using another Bregman divergence. This helps us in finding a better upper bound on error that does not require bounding the subgradient. Note that linearization does not change the nature of analysis. The reason is that we can define $B_f(\theta, \theta_k) = f(\theta) - f(\theta_k) + \langle \nabla f(\theta_k), \theta - \theta_k \rangle$, which means $f(\theta) - B_f(\theta, \theta_k) = f(\theta_k) + \langle \nabla f(\theta_k), \theta - \theta_k \rangle$.

Therefore,

$$\arg \min_{\|\theta - \hat{\theta}_i\|_1^2 \leq R_i^2} \{ \langle \nabla f(\theta_k), \theta - \theta_k \rangle \} = \arg \min_{\|\theta - \hat{\theta}_i\|_1^2 \leq R_i^2} \{ f(\theta) - B_f(\theta, \theta_k) \}.$$

As a result, we can write down the update rule (5a) as

$$\theta_{k+1} = \arg \min_{\|\theta - \hat{\theta}_i\|_1^2 \leq R_i^2} \{ f(\theta) - B_f(\theta, \theta_k) + z_k^T(\theta - y_k) + \rho B_\phi(\theta, Y_k) + \rho_x B_{\phi_x}(\theta, \theta_k) \}.$$

We also have that $B_{\phi_x}(\theta, \theta_k) = B_{\phi'_x}(\theta, \theta_k) - \frac{1}{\rho_x} B_f(\theta, \theta_k)$, which simplifies the update rule to

$$\theta_{k+1} = \arg \min_{\|\theta - \hat{\theta}_i\|_1^2 \leq R_i^2} \{ f(\theta) + \langle z_k, \theta - y_k \rangle + \rho B_\phi(\theta, y_k) + \rho_x B_{\phi_x}(\theta, \theta_k) \}. \quad (26)$$

We notice that equation (26) is equivalent to Equation (7) (Wang and Banerjee, 2013). If in addition we incorporate sampling error, then Lemma 1 (Wang and Banerjee, 2013) changes to

$$\begin{aligned} f(\theta_{k+1}) - f(\hat{\theta}_i) + \lambda_i \|y_{k+1}\|_1 - \lambda_i \|\hat{\theta}_i\|_1 &\leq \\ &- \langle z_k, \theta_{k+1} - y_{k+1} \rangle - \frac{\rho}{2} \{ \|\theta_{k+1} - y_k\|^2 + \|\theta_{k+1} - y_{k+1}\|^2 \} + \langle e_k, \hat{\theta}_i - \theta_k \rangle \\ &+ \frac{\rho}{2} \{ \|\hat{\theta}_i - y_k\|^2 - \|\hat{\theta}_i - y_{k+1}\|^2 \} + \rho_x \{ B_{\phi_x}(\hat{\theta}_i, \theta_k) - B_{\phi_x}(\hat{\theta}_i, \theta_{k+1}) - B_{\phi_x}(\theta_{k+1}, \theta_k) \}. \end{aligned}$$

The above result follows from convexity of f , the update rule for θ (26) and the three point property of Bregman divergence.

By theory of ADMM (Boyd et al., 2011), $\exists B, s.t. \forall z, \|z\|_1 \leq B$. Therefore, $\|z\|_2 \leq B$. Next, following the same approach as in Theorem 4 (Wang and Banerjee, 2013) and considering the sampling error, we get,

$$\begin{aligned} f(\bar{\theta}(T_i)) - f(\hat{\theta}_i) + \lambda_i \|\bar{y}(T_i)\|_1 - \lambda_i \|\hat{\theta}_i\|_1 &\leq \\ \frac{2B^2}{c_2\sqrt{T_i}} + \frac{1}{\sqrt{T_i}} \{ \|\hat{\theta}_i - Y_0\|^2 + c_1 B_{\phi_x}(\hat{\theta}_i, \theta_0) \} &+ \frac{1}{T_i} \sum_{k=1}^{T_i} \langle e_k, \hat{\theta}_i - \theta_k \rangle. \end{aligned}$$

We have $\theta_0 = y_0 = \tilde{\theta}_i$ and $z_0 = 0$. Moreover, $B_{\phi_x}(\theta, \theta_k) = B_{\phi'_x}(\theta, \theta_k) - \frac{1}{\rho_x} B_f(\theta, \theta_k)$. Therefore,

$$\begin{aligned} & f(\bar{\theta}(T_i)) - f(\hat{\theta}_i) + \lambda_i \|\bar{y}(T_i)\|_1 - \lambda_i \|\hat{\theta}_i\|_1 \\ & \leq \frac{2B^2}{c_2 \sqrt{T_i}} + \frac{1}{\sqrt{T_i}} \{ \|\hat{\theta}_i - \tilde{\theta}_i\|^2 + c_1 B_{\phi'_x}(\hat{\theta}_i, \tilde{\theta}_i) - B_f(\hat{\theta}_i, \tilde{\theta}_i) \} + \sum_{k=1}^{T_i} \langle e_k, \hat{\theta}_i - \theta_k \rangle \\ & \leq \frac{2B^2 R_i}{\sqrt{T_i}} + \frac{1}{\sqrt{T_i}} \{ \|\hat{\theta}_i - \tilde{\theta}_i\|^2 + c_1 B_{\phi'_x}(\hat{\theta}_i, \tilde{\theta}_i) \} + \sum_{k=1}^{T_i} \langle e_k, \hat{\theta}_i - \theta_k \rangle. \end{aligned}$$

We note that $\rho_x B_{\phi'_x}(\hat{\theta}_i, \tilde{\theta}_i) = \frac{\rho_x}{2} \|\hat{\theta}_i - \tilde{\theta}_i\|^2$.

Considering the ℓ_2 terms, remember that for any vector x , if $s > r > 0$ then $\|x\|_s \leq \|x\|_r$. Therefore,

$$\frac{\sqrt{\log d}}{R_i} \|\hat{\theta}_i - \tilde{\theta}_i\|_2^2 \leq \frac{\sqrt{\log d}}{R_i} \|\hat{\theta}_i - \tilde{\theta}_i\|_1^2 \leq \frac{\sqrt{\log d}}{R_i} R_i^2 = R_i \sqrt{\log d}.$$

□

C.2 Proof of Proposition 1: Inequality (20a)

Note the shorthand $e_k = \hat{g}_k - \nabla f(\theta_k)$, where \hat{g}_k stands for empirically calculated subgradient of $f(\theta_k)$.

From Lemma 5, we have that

$$f(\bar{\theta}(T_i)) - f(\hat{\theta}_i) + \lambda_i \|\bar{y}(T_i)\|_1 - \lambda_i \|\hat{\theta}_i\|_1 \leq \frac{R_i \sqrt{\log d}}{\sqrt{T_i}} + \frac{R_i B^2}{\sqrt{T_i}} + \frac{R_i^2}{\sqrt{T_i}} + \frac{\sum_{k=1}^{T_i} \langle e_k, \hat{\theta}_i - \theta_k \rangle}{T_i}.$$

Using Lemma 7 from (Agarwal et al., 2012b), we have that

$$\begin{aligned} f(\bar{\theta}(T_i)) - f(\hat{\theta}_i) + \lambda_i \|\bar{y}(T_i)\|_1 - \lambda_i \|\hat{\theta}_i\|_1 & \leq \frac{R_i \sqrt{\log d}}{\sqrt{T_i}} + \frac{R_i B^2}{\sqrt{T_i}} + \frac{R_i^2}{\sqrt{T_i}} + \frac{R_i \sigma_i w_i}{\sqrt{T_i}} \\ & = \frac{R_i \sqrt{\log d}}{\sqrt{T_i}} + \frac{R_i B^2}{\sqrt{T_i}} + \frac{R_i^2}{\sqrt{T_i}} + \frac{R_i \sigma_i}{\sqrt{T_i}} \sqrt{12 \log(3/\delta_i)}. \end{aligned}$$

with probability at least $1 - \delta_i$. In the last equality we use $\delta_i = 3 \exp(-w_i^2/12)$.

C.3 Proof of Lemma 1

Proof follows the same approach as Lemma 1 (Agarwal et al., 2012b). Notice that since we assume exact sparsity the term $\|\theta_{S^c}^*\|_1$ is zero for our case and is thus eliminated. Needless to say, it is an straightforward generalization to consider approximate sparsity from this point.

C.4 Proof of Lemma 2

Using LSC assumption and the fact that $\hat{\theta}_i$ minimizes $f(\cdot) + \|\cdot\|_1$, we have that

$$\begin{aligned} \frac{\gamma}{2} \|\hat{\Delta}(T_i)\|_2^2 & \leq f(\bar{\theta}(T_i)) - f(\hat{\theta}(T_i)) + \lambda_i (\|\bar{y}(T_i)\|_1 - \|\hat{\theta}_i\|_1) \\ & \leq \frac{R_i \sqrt{\log d}}{\sqrt{T_i}} + \frac{R_i B^2}{\sqrt{T_i}} + \frac{R_i^2}{\sqrt{T_i}} + \frac{R_i \sigma_i}{\sqrt{T_i}} \sqrt{12 \log(3/\delta_i)}, \end{aligned}$$

with probability at least $1 - \delta_i$.

C.5 Proof of Proposition 1: Inequality (20b)

Throughout the proof, let $\Delta^*(T_i) = \bar{\theta}_i - \theta^*$ and $\hat{\Delta}(T_i) = \bar{\theta}_i - \hat{\theta}_i$, we have that $\Delta^*(T_i) - \hat{\Delta}(T_i) = \hat{\theta}_i - \theta^*$. Now we want to convert the error bound in (20a) from function values into ℓ_1 and ℓ_2 -norm bounds by exploiting the sparsity of θ^* . Since the error bound in (20a) holds for the minimizer $\hat{\theta}_i$, it also holds for any other feasible vector. In particular, applying it to θ^* leads to,

$$f(\bar{\theta}(T_i)) - f(\theta^*) + \lambda_i \|\bar{y}(T_i)\|_1 - \lambda_i \|\theta^*\|_1 \leq \frac{R_i \sqrt{\log d}}{\sqrt{T_i}} + \frac{R_i B^2}{\sqrt{T_i}} + \frac{R_i^2}{\sqrt{T_i}} + \frac{R_i \sigma_i}{\sqrt{T_i}} \sqrt{12 \log(3/\delta_i)},$$

with probability at least $1 - \delta_i$.

For the next step, we find a lower bound on the left hand side of this inequality.

$$\begin{aligned} f(\bar{\theta}(T_i)) - f(\theta^*) + \lambda_i \|\bar{y}(T_i)\|_1 - \lambda_i \|\theta^*\|_1 &\geq \\ f(\theta^*) - f(\theta^*) + \lambda_i \|\bar{y}(T_i)\|_1 - \lambda_i \|\theta^*\|_1 &= \\ \lambda_i \|\bar{y}(T_i)\|_1 - \lambda_i \|\theta^*\|_1, \end{aligned}$$

where the first inequality results from the fact that θ^* optimizes $f(\theta)$. Thus,

$$\|\bar{y}(T_i)\|_1 \leq \|\theta^*\|_1 + \frac{R_i \sqrt{\log d}}{\lambda_i \sqrt{T_i}} + \frac{R_i B^2}{\lambda_i \sqrt{T_i}} + \frac{R_i^2}{\lambda_i \sqrt{T_i}} + \frac{R_i \sigma_i}{\lambda_i \sqrt{T_i}} \sqrt{12 \log(3/\delta_i)}.$$

Now we need a bound on $\|\bar{\theta}(T_i) - \bar{y}(T_i)\|_1$, we have

$$\|\bar{\theta}(T_i) - \bar{y}(T_i)\|_1 = \left\| \frac{1}{T_i} \sum_{k=1}^{T_i} (\theta_k - y_k) \right\|_1 = \left\| \frac{1}{\tau T_i} \sum_{k=1}^{T_i} (z_{k+1} - z_k) \right\|_1 = \frac{\|z_{T_i}\|_1}{\tau T_i}.$$

By theory of ADMM (Boyd et al., 2011), $\exists B, s.t. \forall z, \|z\|_1 \leq B$.

By triangle inequality

$$\|\bar{\theta}(T_i)\|_1 - \|\bar{y}(T_i)\|_1 \leq \|\bar{\theta}(T_i) - \bar{y}(T_i)\|_1,$$

Hence,

$$\|\bar{\theta}(T_i)\|_1 \leq \|\theta^*\|_1 + \frac{R_i \sqrt{\log d}}{\lambda_i \sqrt{T_i}} + \frac{R_i B^2}{\lambda_i \sqrt{T_i}} + \frac{R_i^2}{\lambda_i \sqrt{T_i}} + \frac{R_i \sigma_i}{\sqrt{T_i}} \sqrt{12 \log(3/\delta_i)} + \frac{B R_i}{T_i \sqrt{T_i}}.$$

By Lemma 6 (Agarwal et al., 2012b),

$$\|\Delta^*(T_i)_{S^c}\|_1 \leq \|\Delta^*(T_i)_S\|_1 + \frac{R_i \sqrt{\log d}}{\lambda_i \sqrt{T_i}} + \frac{R_i B^2}{\lambda_i \sqrt{T_i}} + \frac{R_i^2}{\lambda_i \sqrt{T_i}} + \frac{R_i \sigma_i}{\lambda_i \sqrt{T_i}} \sqrt{12 \log(3/\delta_i)} + \frac{B R_i}{T_i \sqrt{T_i}}.$$

with probability at least $1 - 3 \exp(-w_i^2/12)$.

$\Delta^*(T_i) - \hat{\Delta}(T_i) = \hat{\theta}_i - \theta^*$. Therefore,

$$\begin{aligned} \|\hat{\theta}_i - \theta^*\|_1 &= \\ \|\Delta_S^*(T_i) - \hat{\Delta}_S(T_i)\|_1 + \|\Delta_{S^c}^*(T_i) - \hat{\Delta}_{S^c}(T_i)\|_1 &\geq \\ \{ \|\Delta_S^*(T_i)\|_1 - \|\hat{\Delta}_S(T_i)\|_1 \} - \{ \|\Delta_{S^c}^*(T_i)\|_1 - \|\hat{\Delta}_{S^c}(T_i)\|_1 \}. \end{aligned}$$

Consequently,

$$\|\hat{\Delta}_{S^c}(T_i)\|_1 - \|\hat{\Delta}_S(T_i)\|_1 \leq \|\Delta_{S^c}^*(T_i)\|_1 - \|\Delta_S^*(T_i)\|_1 + \|\hat{\theta}_i - \theta^*\|_1.$$

Using Equation (21b), we get

$$\|\hat{\Delta}_{S^c}(T_i)\|_1 \leq \|\hat{\Delta}_S(T_i)\|_1 + \frac{8s\lambda_i}{\gamma} + \frac{R_i\sqrt{\log d}}{\lambda_i\sqrt{T_i}} + \frac{R_iB^2}{\lambda_i\sqrt{T_i}} + \frac{R_i^2}{\lambda_i\sqrt{T_i}} + \frac{R_i\sigma_i}{\lambda_i\sqrt{T_i}}\sqrt{12\log(3/\delta_i)} + \frac{BR_i}{T_i\sqrt{T_i}}.$$

Hence, further use of the inequality $\|\hat{\Delta}_S(T_i)\|_1 \leq \sqrt{s}\|\hat{\Delta}(T_i)\|_2$ allows us to conclude that there exists a universal constant c such that

$$\|\hat{\Delta}(T_i)\|_1^2 \leq 4s\|\hat{\Delta}(T_i)\|_2^2 + c \left[\frac{s^2\lambda_i^2}{\gamma^2} + \frac{R_i^2\log d}{\lambda_i^2T_i} + \frac{R_i^2B^4}{\lambda_i^2T_i} + \frac{R_i^4}{\lambda_i^2T_i} + \frac{12R_i^2\sigma_i^2\log(3/\delta_i)}{T_i\lambda_i^2} + \frac{B^2R_i^2}{T_i^3} \right], \quad (27)$$

with probability at least $1 - \delta_i$. Optimizing the above bound with choice of λ_i gives us (7). From here on all equations hold with probability at least $1 - \delta_i$, we have

$$\begin{aligned} \|\hat{\Delta}(T_i)\|_1^2 &\leq \frac{8s}{\gamma} \left[f(\bar{\theta}(T_i)) - f(\hat{\theta}(T_i))\lambda_i(\|\bar{Y}(T_i)\|_1 - \|\hat{\theta}_i\|_1) \right] \\ &\quad + \frac{2cs}{\gamma\sqrt{T_i}} \left[R_i\sqrt{\log d} + R_iB^2 + R_i^2 + R_i\sigma_i\sqrt{12\log(3/\delta_i)} \right] + \frac{B^2R_i^2}{T_i^3}. \end{aligned}$$

Thus, for some other c , we have that

$$\|\hat{\Delta}(T_i)\|_1^2 \leq c\frac{s}{\gamma} \left[\frac{R_i\sqrt{\log d}}{\sqrt{T_i}} + \frac{R_iB^2}{\sqrt{T_i}} + \frac{R_i^2}{\sqrt{T_i}} + \frac{R_i\sigma_i}{\sqrt{T_i}}\sqrt{12\log(3/\delta_i)} \right] + \frac{B^2R_i^2}{T_i^3}. \quad (28)$$

Combining the above inequality with error bound (21b) for $\hat{\theta}_i$ and using triangle inequality leads to

$$\begin{aligned} \|\Delta^*(T_i)\|_1^2 &\leq 2\|\hat{\Delta}(T_i)\|_1^2 + 2\|\theta^* - \hat{\theta}_i\|_1^2 \\ &\leq 2\|\hat{\Delta}(T_i)\|_1^2 + \frac{64}{\gamma^2}s^2\lambda_i^2 \\ &\leq c'\frac{s}{\gamma} \left[\frac{R_i\sqrt{\log d}}{\sqrt{T_i}} + \frac{R_iB^2}{\sqrt{T_i}} + \frac{R_i^2}{\sqrt{T_i}} + \frac{R_i\sigma_i}{\sqrt{T_i}}\sqrt{12\log(3/\delta_i)} \right] + \frac{B^2R_i^2}{T_i^3}. \end{aligned}$$

Finally, in order to use $\bar{\theta}(T_i)$ as the next prox center $\tilde{\theta}_{i+1}$, we would also like to control the error $\|\bar{\theta}(T_i) - \hat{\theta}_{i+1}\|_1^2$. Since $\lambda_{i+1} \leq \lambda_i$ by assumption, we obtain the same form of error bound as in (28). We want to run the epoch till all these error terms drop to $R_{i+1}^2 := R_i^2/2$. Therefore, we set the epoch length T_i to ensure that. All above conditions are met if we choose the epoch length

$$T_i = C\frac{s^2}{\gamma^2} \left[\frac{\log d + B^4 + 12\sigma_i^2\log(3/\delta_i)}{R_i^2} + 1 \right],$$

for a suitably large universal constant C . Then, we have that

$$\|\Delta^*(T_i)\|_1^2 \leq \frac{c'}{\sqrt{C}}R_i^2,$$

which completes this proof.

D Proof of Theorem 2

The first step is to ensure that $\|S^* - \tilde{S}_i\|_1^2 \leq R_i^2$, $\|L^* - \tilde{L}_i\|_1^2 \leq \tilde{R}_i^2$ holds at each epoch so that Proposition 2 can be applied in a recursive manner. We prove this in the same manner we proved Theorem 1, by induction on the epoch index. By construction, this bound holds at the first epoch. Assume that it holds for epoch i . Recall that T_i is defined by (13) where $C \geq 1$ is a constant we can choose. By substituting this T_i in inequality (22b), the simplified bound (22b) further yields

$$\|\Delta_S^*(T_i)\|_1^2 \leq \frac{c'}{\sqrt{C}} R_i^2 + c(s+r + \frac{(s+r)^2}{p\gamma^2}) \frac{\alpha^2}{p},$$

Thus, by choosing C sufficiently large, we can ensure that $\|\bar{S}(T_i) - S^*\|_1^2 \leq R_i^2/2 := R_{i+1}^2$. Consequently, if S^* is feasible at epoch i , it stays feasible at epoch $i+1$. Hence, we guaranteed the feasibility of S^* throughout the run of algorithm by induction. As a result, Lemma 3 and 4 apply and for $\tilde{R}_i = c_r R_i$, we find that

$$\|\Delta_S^*(T_i)\|_{\mathbb{F}}^2 \leq \frac{1}{s+r} R_i^2 + (1 + \frac{s+r}{\gamma^2 p}) \frac{2\alpha^2}{p}$$

The bound holds with probability at least $1 - 3\exp(-w_i^2/12)$. The same is true for $\|\Delta_L^*(T_i)\|_{\mathbb{F}}^2$. Recall that $R_i^2 = R_1^2 2^{-(i-1)}$. Since $w_i = w^2 + 24\log i$, we can apply union bound to simplify the error probability as $1 - 6\exp(-w^2/12)$. Let $\delta = 6\exp(-w^2/12)$, we write the bound in terms of δ , using $w^2 = 12\log(6/\delta)$.

Next we convert the error bound from its dependence on the number of epochs k_T to the number of iterations needed to complete k_T epochs, i.e. $T(K) = \sum_{i=1}^K T_i$. Using the same approach as in proof of Theorem 1, we get

$$k_T \simeq -\log \frac{(s+r + (s+r)/\gamma)^2}{R_1^2 T} - \log \left[\log p + B_1^4 + 12\beta^2(p)\sigma^2(\log(6/\delta) + 2\log k_T) + \frac{\gamma B_2 \alpha}{(s+r)\sqrt{p}\tau} \right].$$

As a result

$$\|\Delta_S^*(T_i)\|_{\mathbb{F}}^2 \leq \frac{C(s+r)}{T} \left[\log p + B_1^4 + 12\beta^2(p)\sigma^2(\log(6/\delta) + 2\log k_T) + \frac{\gamma B_2 \alpha}{(s+r)\sqrt{p}\tau} \right] + \frac{\alpha^2}{p} + \frac{B_2 \sqrt{p}}{\tau T}.$$

For the low-rank part, we proved feasibility in proof of Equation (22b), consequently The same bound holds for $\|\Delta_L^*(T_i)\|_{\mathbb{F}}^2$.

D.1 Proofs for Convergence within a Single Epoch for Algorithm 2

We showed that our method is equivalent to running Bregman ADMM on M and $W = [S; L]$. Consequently, our previous analysis for sparse case holds true for the error bound on sum of loss function and regularizers within a single epoch. Hence,

$$\begin{aligned} & f(\bar{M}(T_i)) + \lambda_i \phi(\bar{W}(T_i)) - f(\hat{M}_i) - \lambda_i \phi(\hat{W}(T_i)) \\ & \leq \frac{\|A\hat{W}(T_i) - AW_0\|_{\mathbb{F}}^2}{\sqrt{T_i}} + \frac{c_3 \|\hat{M}(T_i) - M_0\|_{\mathbb{F}}^2}{\sqrt{T_i}} + \frac{B_1^2}{\tau \sqrt{T_i}} + \frac{\sum_{k=1}^{T_i} \text{Tr}(E_k, \hat{M}_i - M_k)}{T_i} \\ & \leq \frac{2(c_3 + 1) \|\hat{S}_i - \tilde{S}_i + \hat{L}_i - \tilde{L}_i\|_{\mathbb{F}}^2}{\sqrt{T_i}} + \frac{B_1^2}{c_4 \sqrt{T_i}} + \frac{\sum_{k=1}^{T_i} \text{Tr}(E_k, \hat{M}_i - M_k)}{T_i} \end{aligned}$$

By the constraints enforced in the algorithm, we have

$$\begin{aligned} & f(\bar{M}(T_i)) + \lambda_i \phi(\bar{W}(T_i)) - f(\hat{M}_i) - \lambda_i \phi(\hat{W}(T_i)) \\ & \leq \sqrt{\frac{R_i^2 + \tilde{R}_i^2}{T_i}} \sqrt{\log p} + \sqrt{\frac{R_i^2 + \tilde{R}_i^2}{T_i}} B_1^2 + \frac{\sum_{k=1}^{T_i} \text{Tr}(E_k, \hat{M}_i - M_k)}{T_i}. \end{aligned}$$

D.2 Proof of Proposition 2: Equation (22a)

In this section we bound the term $\frac{\sum_{k=1}^{T_i} \text{Tr}(E_k, \hat{M}_i - M_k)}{T_i}$. We have

$$M_k - \hat{M}_i = S_k - \hat{S}_i + L_k - \hat{L}_i + (Z_{k+1} - Z_k)/\tau$$

Hence,

$$\begin{aligned} & [\text{Tr}(E_k, \hat{M}_i - M_k)]^2 \\ & \leq [\|E_k\|_\infty \|S_k - \hat{S}_i\|_1 + \|E_k\|_2^2 \|L_k - \hat{L}_i\|_* + \|E_k\|_\infty \|(Z_{k+1} - Z_k)/\tau\|_1]^2 \\ & \leq [2R_i \|E_k\|_\infty + 2\tilde{R}_i \|E_k\|_2 + 2B_1/\tau \|E_k\|_\infty]^2 \\ & \leq \|E_k\|_2^2 [2R_i + 2\tilde{R}_i + 2B_1/\tau]^2 \end{aligned}$$

Consider the term $\|E_k\|_2$. Using Assumption A4, our previous approach in proof of Equation (20a), holds true with addition of a $\beta(p)$ term. Consequently,

$$\begin{aligned} & f(\bar{M}(T_i)) + \lambda_i \phi(\bar{W}(T_i)) - f(\hat{M}_i) - \lambda_i \phi(\hat{W}(T_i)) \leq \\ & \sqrt{\frac{R_i^2 + \tilde{R}_i^2}{T_i}} \sqrt{\log p} + \sqrt{\frac{R_i^2 + \tilde{R}_i^2}{T_i}} B_1^2 + \frac{\beta(p)(R_i + \tilde{R}_i)\sigma_i \sqrt{12 \log(3/\delta_i)}}{\sqrt{T_i}} + \frac{\beta(p)B_1\sigma_i \sqrt{12 \log(3/\delta_i)}}{T_i}. \end{aligned}$$

with probability at least $1 - \delta_i$.

D.3 Proof of Lemma 3

We design R_1 such that $T_i > p$ and our regularizer parameters satisfy

$$\lambda_i \geq 32\rho(\Sigma^*) \sqrt{\frac{\log p}{T_i}} + 4\frac{\alpha}{T_i}, \quad \text{and} \quad \mu_i \geq 16\|\Sigma^*\|_2 \sqrt{\frac{p}{T_i}}. \quad (29)$$

Consequently, we can use Theorem 1 in (Agarwal et al., 2012a) to get

$$\|\hat{S}_i - S^*\|_{\mathbb{F}}^2 + \|\hat{L}_i - L^*\|_{\mathbb{F}}^2 \leq c\{s\frac{\lambda_i^2}{\gamma^2} + r\frac{\mu_i^2}{\gamma^2}\} \quad (30)$$

D.4 Proof of Lemma 4

By LSC condition

$$\begin{aligned} & \frac{\gamma}{2} \|\hat{S}_i - \bar{S}(T_i) + \hat{L}_i - \bar{L}(T_i)\|_{\mathbb{F}}^2 \\ & \leq f(\bar{M}(T_i)) + \lambda_i \phi(\bar{W}(T_i)) - f(\hat{M}_i) - \lambda_i \phi(\hat{W}(T_i)) \\ & \leq \sqrt{\frac{R_i^2 + \tilde{R}_i^2}{T_i}} \sqrt{\log p} + \sqrt{\frac{R_i^2 + \tilde{R}_i^2}{T_i}} B_1^2 + \frac{\beta(p)(R_i + \tilde{R}_i)\sigma_i \sqrt{12 \log(3/\delta_i)}}{\sqrt{T_i}} + \frac{\beta(p)B_1\sigma_i \sqrt{12 \log(3/\delta_i)}}{T_i}, \end{aligned} \quad (31)$$

with probability at least $1 - \delta_i$.

For simplicity, we use

$$H_1 = \sqrt{\frac{R_i^2 + \tilde{R}_i^2}{T_i}} \sqrt{\log p} + \sqrt{\frac{R_i^2 + \tilde{R}_i^2}{T_i}} B_1^2 + \frac{\beta(p)(R_i + \tilde{R}_i)}{\sqrt{T_i}} \sigma_i \sqrt{12 \log(3/\delta_i)} + \frac{\beta(p)B_1 \sigma_i}{T_i} \sqrt{12 \log(3/\delta_i)}.$$

We have,

$$-\frac{\gamma}{2} \text{Tr}(\hat{\Delta}_S \hat{\Delta}_L) = \frac{\gamma}{2} \{\|\hat{\Delta}_S\|_{\mathbb{F}}^2 + \|\hat{\Delta}_L\|_{\mathbb{F}}^2\} - \frac{\gamma}{2} \{\|\hat{\Delta}_S + \hat{\Delta}_L\|_{\mathbb{F}}^2\},$$

In addition,

$$\gamma \|\text{Tr}(\hat{\Delta}_S(T_i) \hat{\Delta}_L(T_i))\| \leq \gamma \|\hat{\Delta}_S(T_i)\|_1 \|\hat{\Delta}_L(T_i)\|_{\infty}.$$

We have,

$$\|\hat{\Delta}_L(T_i)\|_{\infty} \leq \|\hat{L}_i\|_{\infty} + \|\bar{L}(T_i)\|_{\infty}$$

$$\begin{aligned} \|\bar{L}(T_i)\|_{\infty} &\leq \|\bar{Y}(T_i)\|_{\infty} + \|\bar{L}(T_i) - \bar{Y}(T_i)\|_{\infty} \\ &\leq \|\bar{Y}(T_i)\|_{\infty} + \left\| \frac{\sum_{k=0}^{T_i-1} (L_k - Y_k)}{T_i} \right\|_{\infty} \\ &= \|\bar{Y}(T_i)\|_{\infty} + \left\| \frac{\sum_{k=0}^{T_i-1} (U_k - U_{k+1})}{\tau T_i} \right\|_{\infty} \\ &= \|\bar{Y}(T_i)\|_{\infty} + \left\| \frac{-U_{k+1}}{\tau T_i} \right\|_{\infty} \\ &\leq \frac{\alpha}{p} + \frac{B_2}{\tau T_i} \end{aligned}$$

In the last step we incorporated the constraint $\|Y\|_{\infty} \leq \frac{\alpha}{p}$, assumption $\|U\|_{\infty} \leq B_2$ and the fact that $U_0 = 0$. Therefore,

$$\gamma \|\text{Tr}(\hat{\Delta}_S(T_i) \hat{\Delta}_L(T_i))\| \leq \gamma \left(\frac{2\alpha}{p} + \frac{B_2}{\tau T_i} \right) \|\hat{\Delta}_S(T_i)\|_1.$$

Consequently,

$$\frac{\gamma}{2} \|\hat{\Delta}_S(T_i) + \hat{\Delta}_L(T_i)\|_{\mathbb{F}}^2 \geq \frac{\gamma}{2} \{\|\hat{\Delta}_S(T_i)\|_{\mathbb{F}}^2 + \|\hat{\Delta}_L(T_i)\|_{\mathbb{F}}^2\} - \frac{\gamma}{2} \left(\frac{2\alpha}{p} + \frac{B_2}{\tau T_i} \right) \|\hat{\Delta}_S(T_i)\|_1.$$

Combining the above equation with (31), we get

$$\frac{\gamma}{2} \{\|\hat{\Delta}_S(T_i)\|_{\mathbb{F}}^2 + \|\hat{\Delta}_L(T_i)\|_{\mathbb{F}}^2\} - \frac{\gamma}{2} \left(\frac{2\alpha}{p} + \frac{B_2}{\tau T_i} \right) \|\hat{\Delta}_S(T_i)\|_1 \leq H_1.$$

Using $\|S\|_1 \leq \sqrt{p} \|S\|_{\mathbb{F}}$,

$$\begin{aligned} &\|\hat{\Delta}_S(T_i)\|_{\mathbb{F}}^2 + \|\hat{\Delta}_L(T_i)\|_{\mathbb{F}}^2 \\ &\leq \frac{2}{\gamma} \left\{ \sqrt{\frac{R_i^2 + \tilde{R}_i^2}{T_i}} \sqrt{\log p} + \sqrt{\frac{R_i^2 + \tilde{R}_i^2}{T_i}} B^2 + \frac{\beta(p)(R_i + \tilde{R}_i) \sigma_i}{\sqrt{T_i}} \sqrt{12 \log(3/\delta_i)} + \frac{\beta(p)B \sigma_i}{T_i} \sqrt{12 \log(3/\delta_i)} \right\} \\ &\quad + \left(\frac{2\alpha}{\sqrt{p}} + \frac{B_2 \sqrt{p}}{\tau T_i} \right)^2, \end{aligned}$$

with probability at least $1 - \delta_i$.

D.5 Proof of Proposition 2: Equation (22b)

Now we want to convert the error bound in (22a) from function values into vectorized ℓ_1 and Frobenius-norm bounds. Since the error bound in (22a) holds for the minimizer \hat{M}_i , it also holds for any other feasible matrix. In particular, applying it to M^* leads to,

$$\begin{aligned} & f(\bar{M}(T_i)) - f(M^*) + \lambda_i \phi(\bar{W}(T_i)) - \lambda_i \phi(W^*) \\ & \leq \sqrt{\frac{R_i^2 + \tilde{R}_i^2}{T_i}} \sqrt{\log p} + \sqrt{\frac{R_i^2 + \tilde{R}_i^2}{T_i}} B_1^2 + \frac{\beta(p)(R_i + \tilde{R}_i)\sigma_i}{\sqrt{T_i}} \sqrt{12 \log(3/\delta_i)} + \frac{\beta(p)B_1\sigma_i}{T_i} \sqrt{12 \log(3/\delta_i)}, \end{aligned}$$

with probability at least $1 - \delta_i$.

For the next step, we find a lower bound on the left hand side of this inequality.

$$\begin{aligned} & f(\bar{M}(T_i)) - f(M^*) + \lambda_i \phi(\bar{W}(T_i)) - \lambda_i \phi(W^*) \geq \\ & f(M^*) - f(M^*) + \lambda_i \phi(\bar{W}(T_i)) - \lambda_i \phi(W^*) = \\ & \lambda_i \phi(\bar{W}(T_i)) - \lambda_i \phi(W^*), \end{aligned}$$

where the first inequality results from the fact that M^* optimizes M .

From here onward all equations hold with probability at least $1 - \delta_i$. We have

$$\phi(\bar{W}(T_i)) - \phi(W^*) \leq H_1/\lambda_i. \quad (32)$$

i.e.

$$\|\bar{S}(T_i)\|_1 + \frac{\mu_i}{\lambda_i} \|\bar{L}(T_i)\|_* \leq \|S^*\|_1 + \frac{\mu_i}{\lambda_i} \|L^*\|_* + H_1/\lambda_i$$

Using $\bar{S}(T_i) = \Delta_S^* + S^*$, $\bar{L}(T_i) = \Delta_L^* + L^*$. We split Δ_S^* into its on-support and off-support part. We also divide Δ_L^* into its projection onto V and V^\perp . V is range of L^* . Meaning $\forall X \in V, \|X\|_* \leq r$. Therefore,

$$\begin{aligned} & \|(\bar{S}(T_i))_{\text{supp}}\|_1 \geq \|(S^*)_{\text{supp}}\|_1 - \|(\Delta_S^*)_{\text{supp}}\|_1 \\ & \|(\bar{S}(T_i))_{\text{supp}^c}\|_1 \geq -\|(S^*)_{\text{supp}^c}\|_1 + \|(\Delta_S^*)_{\text{supp}^c}\|_1 \\ & \|(\bar{L}(T_i))_V\|_* \geq \|(L^*)_V\|_* - \|(\Delta_L^*)_V\|_* \\ & \|(\bar{L}(T_i))_{V^\perp}\|_* \geq -\|(L^*)_{V^\perp}\|_* + \|(\Delta_L^*)_{V^\perp}\|_* \end{aligned}$$

Consequently,

$$\|(\Delta_S^*)_{\text{supp}^c}\|_1 + \frac{\mu_i}{\lambda_i} \|(\Delta_L^*)_{V^\perp}\|_* \leq \|(\Delta_S^*)_{\text{supp}}\|_1 + \frac{\mu_i}{\lambda_i} \|(\Delta_L^*)_V\|_* + H_1/\lambda_i. \quad (33)$$

$\Delta_S^*(T_i) - \hat{\Delta}_S(T_i) = \hat{S}_i - S^*$. Therefore,

$$\begin{aligned} & \|\hat{S}_i - S^*\|_1 = \\ & \|(\Delta_S^*(T_i))_{\text{supp}} - (\hat{\Delta}_S(T_i))_{\text{supp}}\|_1 + \|(\Delta_S^*(T_i))_{\text{supp}^c} - (\hat{\Delta}_S(T_i))_{\text{supp}^c}\|_1 \geq \\ & \{ \|(\Delta_S^*(T_i))_{\text{supp}}\|_1 - \|(\hat{\Delta}_S(T_i))_{\text{supp}}\|_1 \} - \{ \|(\Delta_S^*(T_i))_{\text{supp}^c}\|_1 - \|(\hat{\Delta}_S(T_i))_{\text{supp}^c}\|_1 \}. \end{aligned}$$

Hence,

$$\|(\hat{\Delta}_S(T_i))_{\text{supp}^c}\|_1 - \|(\hat{\Delta}_S(T_i))_{\text{supp}}\|_1 \leq \|(\Delta_S^*(T_i))_{\text{supp}^c}\|_1 - \|(\Delta_S^*(T_i))_{\text{supp}}\|_1 + \|\hat{S}_i - S^*\|_1.$$

As Equation (29) is satisfied, we can use Lemma 1 (Negahban et al., 2012). Combining the result with Lemma 3, we have $\|\hat{S}_i - S^*\|_1^2 \leq (4s + 3r)(s\frac{\lambda_i^2}{\gamma^2} + r\frac{\mu_i^2}{\gamma^2})$. Consequently, further use of Lemma 3 and the inequality $\|(\hat{\Delta}_S(T_i))_{\text{supp}}\|_1 \leq \sqrt{s}\|\hat{\Delta}(T_i)\|_{\mathbb{F}}$ allows us to conclude that there exists a universal constant c such that

$$\begin{aligned} & \|\hat{\Delta}_S(T_i)\|_1^2 \\ & \leq 4s\|\hat{\Delta}_S(T_i)\|_{\mathbb{F}}^2 + (H_1/\lambda_i)^2 + c(s+r)(s\frac{\lambda_i^2}{\gamma^2} + r\frac{\mu_i^2}{\gamma^2}) + cr\frac{\mu_i^2}{\lambda_i^2} \left[\frac{2}{\gamma}H_1 + \left(\frac{\alpha}{\sqrt{p}} + \frac{B_2\sqrt{p}}{\tau T_i}\right)^2 + s\frac{\lambda_i^2}{\gamma^2} + r\frac{\mu_i^2}{\gamma^2} \right] \\ & \leq 4s \left[\frac{2}{\gamma}H_1 + \left(\frac{\alpha}{\sqrt{p}} + \frac{B_2\sqrt{p}}{\tau T_i}\right)^2 \right] + (H_1/\lambda_i)^2 + c(s+r)(s\frac{\lambda_i^2}{\gamma^2} + r\frac{\mu_i^2}{\gamma^2}) \\ & \quad + cr\frac{\mu_i^2}{\lambda_i^2} \left[\frac{2}{\gamma}H_1 + \left(\frac{\alpha}{\sqrt{p}} + \frac{B_2\sqrt{p}}{\tau T_i}\right)^2 + s\frac{\lambda_i^2}{\gamma^2} + r\frac{\mu_i^2}{\gamma^2} \right] \end{aligned}$$

with probability at least $1 - \delta_i$. Optimizing the above bound with choice of λ_i and complying with the conditions in Lemma 3, leads to

$$\lambda_i^2 = \frac{\gamma}{s+r}H_1 + \frac{\alpha^2}{p^2}$$

Repeating the same calculations for $\|\hat{\Delta}_L(T_i)\|_*$ results in

$$\mu_i^2 = c_\mu \lambda_i^2,$$

we have

$$\|\hat{\Delta}_S(T_i)\|_1^2 \leq c(s+r + \frac{s+r}{\gamma})H_1 + c(s+r)(1 + \frac{s+r}{p\gamma^2})\frac{\alpha^2}{p} + (s+r)(\frac{pB_2^2}{\tau T_i^2} + \frac{B_2\alpha}{\sqrt{p}\tau T_i}).$$

Therefore,

$$\begin{aligned} & \|\Delta_S^*(T_i)\|_1^2 \\ & \leq 2\|\hat{\Delta}_S(T_i)\|_1^2 + 2\|S^* - \hat{S}_i\|_1^2 \\ & \leq 2\|\hat{\Delta}(T_i)\|_1^2 + 8c(s+r)(s\frac{\lambda_i^2}{\gamma^2} + r\frac{\mu_i^2}{\gamma^2}) \\ & \leq c(s+r + \frac{s+r}{\gamma})H_1 + c(s+r)(1 + \frac{s+r}{p\gamma^2})\frac{\alpha^2}{p} + (s+r)(\frac{pB_2^2}{\tau T_i^2} + \frac{B_2\alpha}{\sqrt{p}\tau T_i}) \end{aligned} \tag{34}$$

Finally, in order to use $\bar{S}(T_i)$ as the next prox center \tilde{S}_{i+1} , we would also like to control the error $\|\bar{S}(T_i) - \hat{S}_{i+1}\|_1^2$. Without loss of generality, we can design $\tilde{R}_i = c_r R_i$ for any $0 \leq c_r \leq 1$. The result only changes in a constant factor. Hence, we use $\tilde{R}_i = R_i$. Since $\lambda_{i+1} \leq \lambda_i$ by assumption, we obtain the same form of error bound as in (34). We want to run the epoch till all these error

terms drop to $R_{i+1}^2 := R_i^2/2$. It suffices to set the epoch length T_i to ensure that sum of all terms in (34) is not greater than $R_i^2/2$. All above conditions are met if we choose the epoch length

$$T_i = C \left[\left(s + r + \frac{s+r}{\gamma} \right)^2 \left\{ \frac{\log p + B_1^4 + 12\beta^2(p)\sigma_i^2 \log(6/\delta)}{R_i^2} \right\} + \left(s + r + \frac{s+r}{\gamma} \right) \frac{\beta(p)B_1\sigma_i\sqrt{12\log(6/\delta)}}{R_i^2} \right] \\ + (s+r) \frac{B_2\alpha}{\sqrt{p}\tau R_i^2} + \sqrt{s+r} \frac{\sqrt{p}B_2}{\tau R_i},$$

for a suitably large universal constant C . Then, we have that

$$\|\Delta_S^*(T_i)\|_1^2 \leq \frac{c'}{\sqrt{C}} R_i^2 + c(s+r) \left(1 + \frac{s+r}{p\gamma^2} \right) \frac{\alpha^2}{p}.$$

Since the second part of the upper bound does not shrink in time, we stop where two parts are equal. Namely, $R_i^2 = c(s+r) \left(1 + \frac{s+r}{p\gamma^2} \right) \frac{\alpha^2}{p}$.

With similar analysis for L , we get

$$\|\Delta_L^*(T_i)\|_*^2 \leq \frac{c'}{\sqrt{C}} \frac{1}{1+\gamma} R_i^2 + c \frac{(s+r)^2}{p\gamma^2} \frac{\alpha^2}{p}.$$

E Implementation

Here we discuss the updates for REASON 1 and REASON 2.

Note that for any vector v , $v_{(j)}$ denotes the j -th entry.

E.1 Implementation details for REASON 1

Let us start with REASON 1. We have already provided closed form solution for y and z . The update rule for θ can be written as

$$\begin{aligned} \min_w \quad & \|w - v\|_2^2 \quad s.t. \quad \|w\|_1 \leq R, \\ w = & \theta - \tilde{\theta}_i, \\ R = & R_i, \\ v = & \frac{1}{\rho + \rho_x} \left[y_k - \tilde{\theta}_i - \frac{f(\theta_k)}{\rho} + \frac{z_k}{\rho} + \frac{\rho}{\rho_x} (\theta_k - \tilde{\theta}_i) \right]. \end{aligned} \tag{35}$$

We note that if $\|v\|_1 \leq R$, the answer is $w = v$. Else, the optimal solution is on the boundary of the constraint set and we can replace the inequality constraint with $\|w\|_1 = R$. Similar to (Duchi et al., 2008), we perform Algorithm 3 for solving (35). The complexity of this Algorithm is $\mathcal{O}(d \log d)$, $d = p^2$.

E.2 Implementation details for REASON 2

For REASON 2, the update rule for M , Z, Y and U are in closed form.

Consider the S -update. It can be written in form of (35) with

Algorithm 3 Implementation of θ -update

Input: A vector $v = \frac{1}{\rho + \rho_x}[y_k - \tilde{\theta}_i - \frac{f(\theta_k)}{\rho} + \frac{z_k}{\rho} + \frac{\rho}{\rho_x}(\theta_k - \tilde{\theta}_i)]$ and a scalar $R = R_i > 0$
if $\|v\|_1 \leq R$, **then**
 Output: $\theta = v + \tilde{\theta}_i$
else
 Sort v into μ : $\mu_1 \geq \mu_2 \geq \dots \geq \mu_d$.
 Find $\kappa = \max\{j \in [d] : \mu_j - \frac{1}{j}(\sum_{i=1}^j \mu_i - R) > 0\}$.
 Define $\zeta = \frac{1}{\kappa}(\sum_{i=1}^{\kappa} \mu_i - R)$
 Output: θ , where $\theta_{(j)} = \text{sign}(v_{(j)}) \max\{v_{(j)} - \zeta, 0\} + (\tilde{\theta}_i)_{(j)}$
end if

Algorithm 4 Implementation of S -update

Input: $W^{(1)} = \text{vector}(S_k - \tilde{S}_i)$ and a scalar $R = R_i > 0$
for $t = 1$ to $t = t_s$ **do**
 $v = W^{(t)} - \eta_t \left[\lambda_i \nabla^{(t)} \|W^{(t)}\|_1 + \text{vector}(\tilde{S}_i) + \frac{\rho}{\tau_k} \left(W^{(t)} - \text{vector}(S_k + \tau_k G_{M_k} - \tilde{S}_i) \right) \right]$
 if $\|v\|_1 \leq R$, **then**
 $W^{(t+1)} = v$
 else
 Sort v into μ : $\mu_1 \geq \mu_2 \geq \dots \geq \mu_d$.
 Find $\kappa = \max\{j \in [d] : \mu_j - \frac{1}{j}(\sum_{i=1}^j \mu_i - R) > 0\}$.
 Define $\zeta = \frac{1}{\kappa}(\sum_{i=1}^{\kappa} \mu_i - R)$
 For $1 \leq j \leq d$, $W_{(j)}^{(t+1)} = \text{sign}(v_{(j)}) \max\{v_{(j)} - \zeta, 0\}$
 end if
end for
Output: $\text{matrix}(W^{(t_s)}) + \tilde{S}_i$

$$\begin{aligned} \min_W \quad & \lambda_i \|W + \tilde{S}_i\|_1 + \frac{\rho}{2\tau_k} \|W - (S_k + \tau_k G_{M_k} - \tilde{S}_i)\|_{\mathbb{F}}^2. \quad \text{s.t.} \quad \|W\|_1 \leq R, \\ & W = S - \tilde{S}_i, \quad R = R_i \end{aligned}$$

Therefore, similar to (Duchi et al., 2008), we generate a sequence of $\{W^{(t)}\}_{t=1}^{t_s}$ via

$$W^{(t+1)} = \Pi_1 \left[W^{(t)} - \eta_t \nabla^{(t)} \left(\lambda_i \|W + \tilde{S}_i\|_1 + \frac{\rho}{2\tau_k} \|W - (S_k + \tau_k G_{M_k} - \tilde{S}_i)\|_{\mathbb{F}}^2 \right) \right],$$

where Π_1 is projection on to ℓ_1 norm, similar to Algorithm 3. In other words, at each iteration, vector $\left(W^{(t)} - \eta_t \left[\lambda_i \nabla^{(t)} \|W^{(t)}\|_1 + \tilde{S}_i + \frac{\rho}{\tau_k} (W^{(t)} - (S_k + \tau_k G_{M_k} - \tilde{S}_i)) \right] \right)$ is the input to Algorithm 3 (instead of vector v) and the output is $\text{vector}(W^{(t+1)})$. The term $\nabla^{(t)} \|W^{(t)} + \tilde{S}_i\|_1$ stands for subgradient of the ℓ_1 norm $\|W^{(t)} + \tilde{S}_i\|_1$. The S -update is summarized in Algorithm 4. A step size of $\eta_t \propto 1/\sqrt{t}$ guarantees a convergence rate of $\mathcal{O}(\sqrt{\log p/T})$ (Duchi et al., 2008).

The L -update is very similar in nature to the S -update. The only difference is that the projection is on to nuclear norm instead of ℓ_1 norm. It can be done by performing an SVD before the ℓ_1 norm projection.