STOCHASTIC OPTIMIZATION IN HIGH DIMENSION

by

Hanie Sedghi

A Dissertation Presented to the
FACULTY OF THE USC GRADUATE SCHOOL
UNIVERSITY OF SOUTHERN CALIFORNIA
In Partial Fulfillment of the
Requirements for the Degree
DOCTOR OF PHILOSOPHY
(ELECTRICAL ENGINEERING)
August 2015

To my family

For their endless love, support and encouragement

Acknowledgements

I am truly thankful to my adviser, professor Edmond Jonckheere, who thought me the joy of mathematical research and gave me the freedom of pursuing the problems that I was most interested in and let me grow as a research scientist.

I sincerely thank my co-adviser, professor Anima Anandkumar, who introduced me to the amazing world of theoretical machine learning and helped me conquer new horizons. I am deeply grateful for her dedication, guidance, support and especially her confidence in me. She has been an invaluable mentor for me.

I would like to extend my appreciation to my dissertation committee members, professor Bhaskar Krishnamachari who valued my research and encouraged me and professor Yan Liu for her useful comments.

I am very thankful to Dr. Alekh Agarwal for detailed discussion on his work and his valuable comments.

I am grateful for the fruitful discussions with professor Alex Smola. I have learned a lot from his insightful feedback.

Many thanks to the wonderful staff at Ming Hsieh Department of Electrical Engineering, especially Ms. Diane Demetras, Mr. Tim Boston and Mr. Shane Goodoff.

I would like to extend my greatest thanks to USC graduate school for supporting my PhD studies with USC Provost fellowship and USC graduate school fellowship awards.

A special feeling of gratitude and my warm blessing to the friends who supported me in every respect through this journey, especially Hamed Abrishami, Sanaz Barghi, Hadi Goudarzi, Fatemeh Kashfi, Salman Khaleghi, Mohammad Naghshvar, Parastoo Qarabaqi, Susan Ramirez and Roozbeh Tabrizian.

I wish to acknowledge help and advice from my dear friends and colleagues, Taha Bahadori, Behnam Bahrak, Reza Banirazi, Mohammad Reza Chitgarha, Alireza Imani, Sina Jafarpour, Anantha Karthikeyan, Sina Khaleghi, Mohammad Mehdi Korjani, Phylis LeBourgeois, Sanam Mirzazad, Mohammad Reza Rajati and Katayoun Zand.

Most importantly, I want to thank my family, my source of inspiration, courage and joy. My husband, Majid Janzamin who spurred me on with his patience, enthusiasm and abundant love and lit my path with hope and peace past all obstacles; my parents, Mitra Soraya and Kambiz Sedghi who raised me up with their heavenly love, sacrifice, wisdom and devotion to become who I am, embraced my fears and gave me the strength to never surrender; and my wonderful sisters, Hosna and

Hoda who stood by me through all life challenges and without their unconditional love, passion and encouragement, I could not chase my dreams.

I am deeply grateful to my gracious aunt Fatemeh, my loving grandmother Effat and my kind uncle, Hossein whose pure love, compassion and sincere prayers have accompanied me in every stage of my life. I am thankful to my brother-in-law, Mehdi for his support and kindness through these years.

Abstract

In this thesis, we consider two main problems in learning with big data: data integrity and high dimension. We specifically consider the problem of data integrity in smart grid as it is of paramount importance for grid maintenance and control. In addition, data manipulation can lead to catastrophic events. Inspired by this problem, we then expand the horizon to designing a general framework for stochastic optimization in high dimension for any loss function and any underlying low dimensional structure. We propose Regularized Epoch-based Admm for Stochastic Optimization in high-dimensioN (REASON). Our ADMM method is based on epoch-based annealing and consists of inexpensive steps which involve projections on to simple norm balls. We provide explicit bounds for the sparse optimization problem and the noisy matrix decomposition problem and show that our convergence rate in both cases matches the minimax lower bound. For matrix decomposition into sparse and low rank components, we provide the first guarantees for any online method. Experiments show that for both sparse optimization and matrix decomposition problems, our algorithm outperforms the state-of-the-art methods. In particular, we reach higher accuracy with same time complexity.

Table of Contents

Appendix

List Of Tables

List Of Figures

Chapter 1

Introduction

With new trend in technology, there is the challenge of data deluge in almost any domain. We have lots of data, in terms of system measurement, image, video, genetics, social network, etc and the amount just increases. The goal is to use the data for inference and even control. But as we have more data it does not mean we have more information. Data might have corruptions. Therefore, ensuring data integrity is an important task.

In addition, with more data comes more challenges. Although we have more data, we have a huge number of unknown parameters. The classic approach in statistics is that with fixed number of parameters p, the sample size $n \to \infty$. But in modern applications in science and engineering we have large scale problems where both n, p can be very large (possibly $p \gg n$) and this calls for high-dimensional theory where we let both $(n, p) \to \infty$. In high-dimensional statistics we have exponential explosions in computational complexity and also for a large number of unknown parameters the sample complexity goes beyond the number of available samples. Therefore, for a tractable analysis of the problem with available samples, additional assumptions are made on the problem structure, i.e., an embedded low dimensional structure. Examples include sparse vectors, patterned matrices, low rank matrices, Markov random fields and some assumptions on manifold structure. Nevertheless, even within the class of tractable problems, high-dimensional optimization is harmed by curse of dimensionality. To be more specific, conventional optimization methods provide convergence bounds that are a quadratic function of the dimension. Therefore, their convergence guarantees in high dimension are disheartening.

To make the challenge even more severe, we should consider another challenge in big data. That is the volume and velocity of the data we receive, which calls for learning methods that are fast, cheap and do not require data storage. Therefore, batch methods are not a good fit for such applications and we need stochastic methods that can provide a good estimation of the parameter per any noisy sample they receive. Therefore, we no more have the luxury of noise concentration that batch optimization benefits from. Hence, the combination of stochastic optimization and high-dimensional statistics leads to a extremely difficult problem.

In this thesis we consider the challenges of learning with big data. To be more precise, our goal is to shed light on the problem of big data from two angles. First we consider the problem of data integrity. This is motivated by the fact that not all the data we receive is reliable and for some cases believing the data without any integrity check can lead to catastrophic events. One prominent case is the case of data integrity in smart grid. We elaborate on this shortly.

Inspired by the complex problem of data integrity in smart grid, we used it as our spring board to extend our horizon and consider the general case of stochastic optimization in high dimension, for any loss function with some mild assumptions. Hence, our approach can be used for various applications. Our goal is to design a general method for stochastic optimization method in high-dimensional setting that is fast and cheap to implement and to provide tight convergence guarantees that have logarithmic dependence with dimension (this is the minimax optimal rate). We also compare our method with earlier state-of-the-art methods via experiments.

1.1 Initial Results: Data integrity in Smart Grid

Recently, Gaussian graphical models have been used as a precious tool for modeling and analyzing various phenomena in diverse fields such as control, cyber security, biology, sociology, social networks and geology. It all starts with model selection for the random variables associated to the problem. Model selection means finding the real underlying Markov graph among a group of random variables based on samples of those random variables.

Traditionally, the term *grid* is used to refer to an electricity system that supports the following four operations: electricity generation, electricity transmission, electricity distribution, and voltage stability control. In the early days, generation was co-located with distribution in what we would now call a *micro-grid* and the connections among the micro-grids were meant to transmit energy in case of such contingencies as shift in the supply/demand balance. After deregulation, however, a large-scale generation-transmission-distribution *network* became the substitute for the traditional generation-distribution *co-location.* The new network allows consumers to purchase electricity at the cheapest price across the country, as opposed to the former concept in which consumers were forced to purchase electricity from local utility companies. Other considerations calling for an overhaul of the electricity system include the reduction of carbon emission, an objective that cannot be achieved without a significant contribution from the electricity sector. This calls for a bigger share of the renewable energy resources in the generation mix and a supply/demand that must be managed more effectively. Management and control of the grid made increasingly complex by its response to electricity market conditions are, next to its ability to detect contingencies, the most fundamental attributes that make it *smart.*

Automated large scale management requires considerable exchange of information, so that the smart grid has become a two-commodity flow—electricity and information. By utilizing modern information technologies, the smart grid is capable of delivering power in a more efficient way and responding to wider ranging conditions.

Massive amount of measurements and their transmission across the grid by modern information technology, however, make the grid prone to attacks. Fast and accurate detection of possibly malicious events is of paramount importance not only for preventing faults that may lead to blackouts, but also for routine monitoring and control tasks of the smart grid, including state estimation and optimal power flow. Fault localization in the nation's power grid networks is known to be challenging, due to the massive scale and inherent complexity.

We use model selection to address this crucial matter for smart grid control and maintenance. We approach false data injection in smart grid via statistical analysis of underlying structure among data. In order to learn the structure of the power grid, we utilize the new Gaussian Graphical Model Selection method called Conditional Covariance Test (CMIT) [Anandkumar et al., 2012] and prove that in normal conditions this structure follows grid topology. Next we assess that our method can detect a sophisticated and strong attack as it causes the graphical model to change. Our approach is the first method that can comprehensively detect this data manipulation without the need for additional hardware.

1.2 Main Results: Stochastic Optimization in High Dimension

Stochastic optimization considers the problem of minimizing a loss function with access to noisy samples of (gradient of) the function. The goal is to have an estimate of the optimal parameter (minimizer) per new sample. Therefore, compared to batch optimization where we have noise concentration, stochastic optimization is a more challenging problem.

In addition, as discussed earlier, in high dimensional statistics we have $p \gg n$. Therefore in general the problems are not tractable. in order to make the problem tractable we need to assume low dimensional underlying truth such as sparse vectors, patterned matrices, low rank matrices, Markov random fields and some assumptions on manifold structure. Mathematically this is modeled by adding a regularizer term to the optimization problem. These terms are mostly non-smooth and hence, make each step of stochastic optimization very expensive. The reason is that in most cases no closed form solution exists for each step of the optimization problem.

The alternating direction method of multipliers (ADMM), takes the form of a decomposition-coordination procedure, in which the solutions to small local subproblems are coordinated to find a solution to a large global problem. To be more precise, ADMM decomposes the optimization problem into two parts; minimizing the loss function term and minimizing the regularization term and then links the two solutions. ADMM can be viewed as an attempt to blend the benefits of dual decomposition and augmented Lagrangian methods for constrained optimization. It is a simple but powerful algorithm that is well suited to distributed convex optimization, and in particular to problems arising in applied statistics and machine learning. Nevertheless, stochastic ADMM techniques suffer from the curse of dimensionality, i.e., their convergence rates are proportional to square of the dimension which is disheartening for high-dimensional problem.

In order to design a general framework for stochastic optimization in high-dimension that is both fast and cheap as well as enjoys logarithmic dependence to dimension (and is hence minimax optimal), modify stochastic ADMM such that we have best of both worlds. We do this an epoch-based approach and by performing intelligent projections into a norm ball around the optimal value and shrink the ball after each epoch such that we have error contraction by the end of each epoch. The norm ball is determined by the nature of the underlying optimal value or hence the regularizer term. For example, in case of sparse optimization we use ℓ_1 norm projections. It should be noted that we do not have a knowledge of the optimal parameter and hence use the average of the last epoch estimates as an estimate of the optimal value. We design our algorithm parameters such that we ensure the optimal parameter remains feasible during the algorithm. In addition, we prove that by our choice of parameters, the square error shrinks by half by the end of each epoch. Therefore, we halve the radius of the norm ball and can obtain a logarithmic dependency to the dimension. This is a general framework that can be applied to any convex optimization problem with some mild assumptions and any number of regularizers. We provide complete detailed analysis for two infamous problems: sparse optimization and matrix decomposition into sparse and low rank parts (also known as robust PCA).

The above simple modifications to ADMM have huge implications for high-dimensional problems. For sparse optimization, our convergence rate is $\mathcal{O}(\frac{s \log d}{T})$, for s-sparse problems in d dimensions in T steps. Our bound has the best of both worlds: efficient high-dimensional scaling (as $\log d$) and efficient convergence rate (as $\frac{1}{T}$). This also matches the minimax lower bound for the linear model and square loss function [Raskutti et al., 2011], which implies that our guarantee is unimprovable by any (batch or online) algorithm (up to constant factors). For matrix decomposition, our convergence rate is $\mathcal{O}((s+r)\beta^2(p)\log p/T)) + \mathcal{O}(\max\{s+r, p\}/p^2)$ for a $p \times p$ input matrix in T steps, where the sparse part has s non-zero entries and low rank part has rank r. For many natural noise models (e.g. independent noise, linear Bayesian networks), $\beta^2(p) = p$, and the resulting convergence rate is minimax-optimal. Note that our bound is not only on the reconstruction error, but also on the error in recovering the sparse and low rank components. These are the first convergence guarantees for online matrix decomposition in high dimensions. Moreover, our convergence rate holds *with high probability* when noisy samples are input, in contrast to expected convergence rate, typically analyzed in literature. See Table 4.1, 4.2 for comparison of this work with related frameworks.

Our proposed algorithms provide significantly faster convergence in high dimension and better robustness to noise. For sparse optimization, our method has significantly better accuracy compared to the stochastic ADMM method and better performance than RADAR, based on multi-step dual averaging [Agarwal et al., 2012b]. For matrix decomposition, we compare our method with the state-of-art inexact ALM [Lin et al., 2010] method. While both methods have similar reconstruction performance, our method has significantly better accuracy in recovering the sparse and low rank components.

Chapter 2

Preliminaries

2.1 Notation and Terminology

1. A graph G is represented as $G(V, E)$ where V represents the set of vertices and E represents the edge set.

2. For random variables $\perp$ is used to show independence and $|$ symbol is used for conditioning. i.e., $X_1 \perp X_2 | X_3$ means X_1 is independent of X_2 given X_3.

3. In probability theory and statistics, a covariance matrix is a matrix whose element in the i, j position is the covariance between the i th and j th elements of a random vector (that is, of a vector of random variables).

4. For every matrix A, $\mathrm{Tr}(A)$ represents sum of the diagonal entries of A.

In the sequel, we use lower case letter for vectors and upper case letter for matrices.$\|x\|_1$, $\|x\|_2$ refer to ℓ_1, ℓ_2 vector norms respectively. The term $\|X\|_*$ stands for nuclear norm of X. In addition, $\|X\|_2$, $\|X\|_{\mathbb{F}}$ denote spectral and Frobenius norms respectively. $|||X|||_\infty$ stands for induced infinity norm. We use vectorized ℓ_1, ℓ_∞ norm for matrices. i.e., $\|X\|_1 = \sum_{i,j} |X_{ij}|$, $\|X\|_\infty = \max_{i,j} |X_{ij}|$.

2.2 High Dimensional Statistics

The field of high-dimensional statistics studies data whose dimension is higher than the dimension of classical multivariate data. In many applications the dimension of the data is bigger than the sample size. High-dimensional statistical inference deals with models in which the the number of parameters p is comparable to or larger than the sample size n. Since it is usually impossible to obtain consistent procedures unless $p/n \to 0$, Therefore, for a tractable analysis of the problem with available samples, a line of recent work has studied models where additional assumptions are made on the problem structure, i.e., an embedded low dimensional structure. with various types of low-dimensional structure, including sparse vectors, patterned matrices, low rank matrices, Markov random fields and some assumptions on manifold structure. In such settings, a general approach to estimation is to solve a regularized optimization problem, which combines a loss function measuring how well the model fits the data with some regularization function that encourages the assumed structure. Accordingly, there are now several lines of work within high-dimensional statistics, all of which are based on imposing some type of low-dimensional constraint on the model space and then studying the behavior of different estimators. Examples include linear regression with sparsity constraints, estimation of structured covariance or inverse covariance matrices, graphical model selection, sparse principal component analysis, low-rank matrix estimation, matrix decomposition problems and estimation of sparse additive nonparametric models. The classical technique of regularization has proven fruitful in all of these contexts. Many well-known estimators are based on solving a convex optimization problem formed by the sum of a loss function with a weighted regularizer. For example, ℓ_0 norm denotes the number of nonzero elements but since ℓ_0 is a nonconvex function, ℓ_1 norm is used to ensure sparsity. The reason is that ℓ_1 norm is the closest function to ℓ_0 that is convex. As another example, to ensure low rank structure nuclear norm is used as a regularizer. The intuition is that nuclear norm is sum of the singular values and minimizing the nuclear norm results in imposing the low rank structure.

2.3 Graphical Model

Definition 1. ***Global Markov property***: *A probability distribution is said to have* global Markov property *with respect to a graph if, for any disjoint subsets of nodes I, J, S such that S separates I and J on the graph, the distribution satisfies $X_I \perp X_J | X_S$, i.e., X_I is independent of X_J conditioned on X_S. This is represented in Figure 2.1.*

Figure 2.1: Global Markov property: $X_I \perp X_J | X_S$

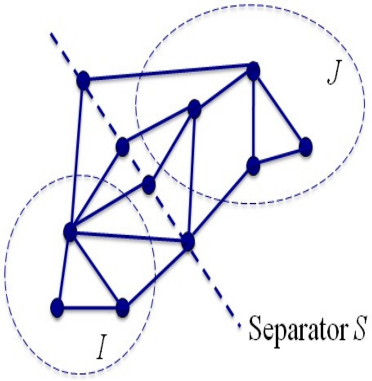
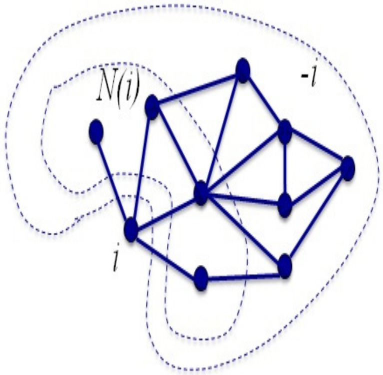


Figure 2.2: Local Markov property: $X_i \perp X_{\mathcal{V}\setminus\{i \cup N(i)\}} | X_{N(i)}$



Definition 2. ***Pairwise Markov property***: *A distribution is* pairwise Markov *with respect to a given graph if, for any two nodes i and j in the graph such that there is no direct link in the graph between i and j, then X_i is independent of X_j given the states of all of the remaining nodes, i.e., $X_i \perp X_j | X_{\mathcal{V}\setminus\{i,j\}}$.*

Definition 3. ***Local Markov property***: *A set of random variables is said to have* local Markov property *corresponding to a graph [Lauritzen, 1996] if any variable X_i is conditionally independent of all other variables $X_{\mathcal{V}\setminus\{i \cup N(i)\}}$ given its neighbors*

$X_{N(i)}$, where $\mathcal{V} \setminus \{i \cup N(i)\} := \{j \in \mathcal{V} : j \neq i, j \neq N(i)\}$ and $N(i) := \{j \in \mathcal{V} : (i,j) \in \mathcal{E}\}$. Local Markov property can be seen in Figure 2.2.

Definition 4. ***Markov Random Field (MRF)****:Given an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, a set of random variables $X = (X_v)_{v \in \mathcal{V}}$ form a* Markov Random Field *with respect to $\mathcal{G}$ if they have the global Markov property. It should be noted that local Markov property and pairwise Markov property are equivalent and they are a special case of global Markov property. For a strictly positive probability distribution, the properties are equivalent and it can be shown that the probability distribution can be factorized with respect to the graph [Lauritzen, 1996].*

One instance of this positivity condition happens in case of jointly Gaussian distributions.

Definition 5. ***Gaussian Markov Random Field (GMRF)****: A Gaussian Markov Random Field (GMRF) is a family of jointly Gaussian distributions, which factor according to a given graph. Given a graph $G = (V, E)$, with $V = \{1, ..., p\}$, consider a vector of Gaussian random variables $X = [X_1, X_2, ..., X_p]^T$, where each node $i \in V$ is associated with a scalar Gaussian random variable X_i. A Gaussian Markov Random Field on G has a probability density function (pdf) that may be parametrized as*

$$f_X(x) \propto exp[-\frac{1}{2}x^T Jx + h^T x]; \qquad (2.3.1)$$

where J is a positive-definite symmetric matrix whose sparsity pattern corresponds to that of the graph G. More precisely,

$$J(i,j) = 0 \Longleftrightarrow (i,j) \notin E. \tag{2.3.2}$$

The matrix $J = \Sigma^{-1}$ is known as the ***potential*** *or* ***information*** *matrix, the non-zero entries $J(i,j)$ as the edge potentials, and the vector h as the vertex potential vector.*

Definition 6. ***Graphical Model****: In general, Graph $G = (V, E)$ is called the Markov graph (graphical model) underlying the joint probability distribution $f_X(x)$ where the node set V represents each random variable X_i and the edge set E is defined in order to satisfy local Markov property. For a Markov Random Field, local Markov property states that $X_i \perp X_{-\{i,N(i)\}}|X_{N(i)}$, where $X_{N(i)}$ represents all random variables associated with the neighbors of i in graph G and $X_{-\{i,N(i)\}}$ denotes all variables except for X_i and $X_{N(i)}$.*

2.3.1 KullbackLeibler divergence

Definition 7. ***KullbackLeibler divergence****: In probability theory and information theory, the KullbackLeibler divergence [Kullback, 1951, 1959, 1987] (also information divergence, information gain, relative entropy, or KLIC) is a non-symmetric measure of the difference between two probability distributions P and Q. Specifically,* *the KullbackLeibler divergence of Q from P, denoted $DKL(P \parallel Q)$, is a measure of the information lost when Q is used to approximate P [K. P. Burnham, 2002]. KL measures the expected number of extra bits required to code samples from P when using a code based on Q, rather than using a code based on P. Typically P represents the "true" distribution of data, observations, or a precisely calculated theoretical distribution. The measure Q typically represents a theory, model, description, or approximation of P.*

For discrete probability distributions P and Q, the KL divergence of Q from P is defined to be

$$D_{KL}(P \parallel Q) = \sum_i \ln\left(\frac{P(i)}{Q(i)}\right) \qquad (2.3.3)$$

In words, it is the expectation of the logarithmic difference between the probabilities P and Q, where the expectation is taken using the probabilities P.

For distributions P and Q of a continuous random variable, KL-divergence is defined to be the integral [Bishop, 2006]

$$D_{KL}(P \parallel Q) = \int_{-\infty}^{\infty} \ln\left(\frac{P(x)}{Q(x)}\right) p(x)dx \qquad (2.3.4)$$

where p and q denote the densities of P and Q.

In Bayesian statistics the KL divergence can be used as a measure of the information gain in moving from a prior distribution to a posterior distribution. If some new fact $Y = y$ is discovered, it can be used to update the probability distribution for X from $p(x|I)$ to a new posterior probability distribution $p(x|y, I)$ using Bayes' theorem:

$$p(x|y, I) = \frac{p(y|x, I)p(x|I)}{p(y|I)}. \tag{2.3.5}$$

This distribution has a new entropy

$$H(p(.|y, I)) = \sum_x p(x|y, I) \log p(x|y, I), \tag{2.3.6}$$

which may be less than or greater than the original entropy $H(p(|I))$. However, from the standpoint of the new probability distribution one can estimate that to have used the original code based on $p(x|I)$ instead of a new code based on $p(x|y, I)$ would have added an expected number of bits

$$D_{KL}(p(.|y, I) \parallel p(.|I)) = \sum_x p(x|y, I) \frac{\log p(x|y, I)}{p(x|I)} \tag{2.3.7}$$

to the message length. This therefore represents the amount of useful information, or information gain, about X, that we can estimate has been learned by discovering $Y = y$.

2.4 Convex Optimization

A convex optimization problem is one of the form

$$\text{minimize} \;\; f_0(x)$$
$$\text{subject to} \;\; f_i(x) \leq b_i, i = 1, ..., m,$$

where the functions $f_0, ..., f_m : \mathbb{R}^n \to \mathbb{R}$ are convex, i.e., satisfy $f_i(\alpha x + \beta y) \leq \alpha f_i(x) + \beta f_i(y)$ for all $x, y \in \mathbb{R}^n$ and all $\alpha, \beta \in \mathbb{R}$ with $\alpha + \beta = 1, \alpha \geq 0, \beta \geq 0$.

Convex optimization enjoys the fact that for a convex optimization there is no spurious local optima, i.e., the local optima is the global optima. In addition, convex optimization problems can be solved efficiently.

2.4.1 Stochastic Optimization

Stochastic optimization considers the problem of minimizing a loss function with access to noisy samples of (gradient of) the function. The goal is to have an estimate of the optimal parameter (minimizer) per new sample.

Consider the optimization problem

$$\theta^* \in \arg\min_{\theta \in \Omega} \mathbb{E}[f(\theta, x)], \tag{2.4.1}$$

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

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

is the batch optimal solution.

2.4.2 Alternating Direction Method of Multipliers

Alternating Direction Method of Multipliers (ADMM) is an algorithm that is intended to blend the decomposability of dual ascent with the superior convergence properties of the method of multipliers. The algorithm solves problems in the form

$$\text{minimize} \quad f(x) + g(y) \quad \text{subject to} \quad Ax + By = c$$

with variables $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^m$, where $A \in \mathbb{R}^{p \times n}, B \in \mathbb{R}^{p \times m}$, and $c \in \mathbb{R}^p$. We will assume that f and g are convex. The difference from the usual convex optimization problem is that the variable x is split into two parts and the cost function is separable. As in the method of multipliers, we form the augmented Lagrangian

$$\mathcal{L}_\rho(x, y, z) = f(x) + g(y) + z^\top(Ax + By - c) + \frac{\rho}{2}\|Ax + By - c\|^2.$$

ADMM iterations are the as follows:

$$\begin{aligned}
x_{k+1} &:= \arg\min_x \; \mathcal{L}_\rho(x, y_k, z_k), \\
y_{k+1} &:= \arg\min_y \; \mathcal{L}_\rho(x_{k+1}, y, z_k), \\
z_{k+1} &:= z_k + \rho(Ax_{k+1} + By_{k+1} - c),
\end{aligned}$$

where $\rho > 0$. It is proved that for convex, closed and proper functions f, g ADMM converges [Boyd et al., 2011].

ADMM is originally a batch method. However, with some modifications it can also be used for stochastic optimization. Since in stochastic setting we only have access to noisy samples of gradient, we use an inexact approximation of the Lagrangian as

$$\begin{aligned}
\hat{\mathcal{L}}_{\rho,k} = {} & f_1(x_k) + \langle \nabla f(x_k, \zeta_{k+1}), x\rangle + g(y) - z^\top(Ax + By - c) \\
& + \frac{\rho}{2}\|Ax + By - c\|^2 + \frac{\|x - x_k\|^2}{2\eta_{k+1}},
\end{aligned}$$

where η_{k+1} is a time-varying step size [Ouyang et al., 2013]. Note that the first term does not appear in ADMM iterates. Convergence rate for stochastic ADMM is shown in table 4.1. It can be seen that the convergence rate is proportional to square of dimension which is a disheartening rate in high dimension.

As discussed above, in stochastic ADMM we use noisy samples of gradient. Therefore, in order to prove convergence for stochastic ADMM, we need a bound on gradient, i.e., the cost function needs to satisfy an additional assumption: Lipschitz property.

Definition 8. ***Lipschitz property:*** *A function $f : \Omega \to \mathbb{R}$ is said to satisfy the Lipschitz condition if there is a constant M such that*

$$|f(x) - f(x')| \leq M\|x - x'\| \quad \forall\ x, x' \in \Omega. \tag{2.4.2}$$

The smallest constant M satisfying (2.4.2) *is called Lipschitz constant. Lipschitz constant can be interpreted as an upper bound on gradient of function f.*

Another property that improves the convergence rate is strong convexity.

Definition 9. ***Strong Convexity:*** *A differentiable function $f : \Omega \to \mathbb{R}$ is called strongly convex if there is a constant $m > 0$ such that*

$$f(x') \geq f(x) + \langle \nabla f(x), x' - x \rangle + \frac{m}{2}\|x' - x\|^2 \quad \forall\ x, x' \in \Omega.$$

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.

In high dimension, we cannot guarantee the above properties globally. Nevertheless we will show that the following locally defined notions suffice.

Definition 10. ***Local Lipschitz condition:*** *For each $R > 0$, there is a constant $G = G(R)$ such that*

$$|f(\theta_1) - f(\theta_2)| \leq G\|\theta_1 - \theta_2\|_1$$

for all $\theta_1, \theta_2 \in S$ such that $\|\theta - \theta^*\|_1 \leq R$ and $\|\theta_1 - \theta^*\|_1 \leq R$.

Definition 11. ***Local strong convexity (LSC)****: The function $f : S \to \mathbb{R}$ 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 $\|\theta_2\|_1 \leq R$.

Chapter 3

Initial Results: Data Integrity in Smart Grid

3.1 Introduction

Among the attributes that make the grid "smart" is its ability to process a massive amount of data for monitoring, control, and maintenance purposes. In a typical Transmission System Operator (TSO), the substation Remote Terminal Units (RTUs) read the status of voltages, currents, and switching states. The RTU data is redirected in data-packages to the Supervisory Control and Data Acquisition (SCADA) system via communication channels. In addition, synchronous Phasor Measurement Units (PMUs) are being massively deployed throughout the grid. PMUs provide a higher level of detail to the SCADA system (e.g. voltage angle). The signals from the PMUs are transmitted via the RTU to the SCADA. The State Estimator (SE) located at the control center aims to find the *best* overall snapshot solution based on *all* measurements.

Recent monitoring and control schemes rely primarily on PMU measurements; for example, [Diao et al., May 2009] tries to increase voltage resilience to avoid voltage collapse by using synchronized PMU measurements and decision trees and [Zhu and Giannakis, C. Wei, 2012, He and Zhang, June 2011] rely on PMUs for fault detection and localization.

The centralization of the data to the State Estimator makes it the back door to false data injection attacks. Therefore, aforementioned methods can be deluded by false data injection attacks. Thus, it is crucial to have a mechanism for fast and accurate discovery of malicious tampering; both for preventing the attacks that may lead to blackouts, and for routine monitoring and control tasks of the smart grid. The cyber attacks have gained increasing attention over the past years. Unfortunately, there are realistic "stealthy" threats that cannot be detected with current security modules in the power network and may lead to cascading events, instability in the system, and blackouts in major areas of the network. For details on stealthy deception attack, their implementation and serious consequences, see [Kwon et al., 2013, Kosut et al., 2010, Amin and Giacomoni, 2012, Giani et al., January 2012, Yao Liu and Reiter, May 2011].

3.1.1 Summary of Results

We have designed a decentralized false data injection attack detection mechanism that utilizes the Markov graph of the bus phase angles. We utilize the conditional

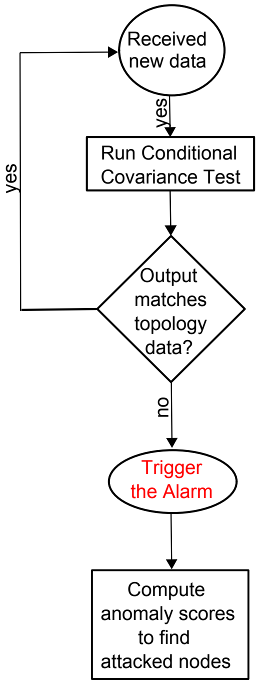

Received new data
yes
Run Conditional Covariance Test
yes
Output matches topology data?
no
Trigger the Alarm
Compute anomaly scores to find attacked nodes

Figure 3.1: Flowchart of our detection algorithm

covariance threshold test CMIT [Anandkumar et al., 2012] to learn the structure of the grid. We show that under normal circumstances, and because of the grid structure, the Markov graph of voltage angles can be determined by the power grid graph. Therefore, a discrepancy between calculated Markov graph and learned structure triggers the alarm. This work was initiated by the authors in [Sedghi and Jonckheere, 2013].

Because of the connection between the Markov graph of the bus angle measurements and the grid topology, our method can be implemented in a decentralized manner, i.e. at each sub-network. Currently, sub-network topology is available online and global network structure is available hourly [Zhu and Giannakis]. Not only by decentralization can we increase the speed and get closer to online detection, but we also increase accuracy and stability by avoiding communication delays and synchronization problems when trying to send measurement data between locations far apart [Zhu et al., 2013, Ancillotti et al., 2013]. Furthermore, we noticeably decrease the amount of exchanged data to address privacy concerns as much as possible.

We show that our method can detect the most recently designed attack on the power grid that remains undetected by the traditional bad data detection scheme [Teixeira et al., 2011] and is capable of deceiving the State Estimator and damaging power network control, monitoring, demand response, and pricing schemes [Kosut et al., 2010]. In this scenario, the attacker is equipped with vital data and has the knowledge of the bus-branch model of the grid. It should be noted that our method not only detects that the system is under attack, but also determines the particular set of nodes under the attack. The flowchart is shown in Figure 3.1.

In addition, we show that our method can detect the situation where the attacker manipulates reactive power data to lead the State Estimator to wrong estimates of the voltages. Such an attack can be designed to fake a voltage collapse or trick the operator to cause a voltage collapse. This latter detection is based on the linearization of the AC power flow around the steady state. Then using our algorithm for bus voltages and reactive power rather than bus phase angles and active power, it readily follows that this latter attack can also be detected.

3.1.2 Related Work

Although the authors of [Giani et al., January 2012] suggest an algorithm for PMU placement such that the "stealthy" attack is observable, they report a successful algorithm only for the 2-node attack and propose empirical approaches for the 3, 4, and 5-node attacks. According to [Giani et al., January 2012], for cases where more than two nodes are under attack, the complexity of the approach is said to be *"disheartening"*. Considering the fact that finding the number of needed PMUs is NP-hard and that [Giani et al., January 2012] gives an upper bound and uses a heuristic method for PMU placement, we need to mention for comparison purposes that our algorithm has no hardware requirements, its complexity does not depend on the number of nodes under attack, and it works for any number of attacked nodes. It is also worth mentioning that, even in the original paper presenting the attack for a relatively small network (IEEE-30), seven measurements from five nodes are manipulated. Therefore, it seems that the 2-node attack is not the most probable one.

There has been another line of work dedicated to computing the "security index" for different nodes in order to find the set of nodes that are most vulnerable to false data injection attacks [Hendrickx et al., 2014]. Although these attempts are acknowledged, our method differs greatly from such perspectives as such methods do not detect the attack state when it happens and they cannot find the set of nodes that are under attack.

The dependency graph approach is used in [He and Zhang, June 2011] for topology fault detection in the grid. However, since attacks on the State Estimator are not considered, such methods can be deceived by false data injection. Furthermore, [He and Zhang, June 2011] use a constrained maximum likelihood optimization for finding the information matrix, while here an advanced structure learning method is used that captures the power grid structure better. This is because in the power grid the edges are not centered but distributed all over the network. This is discussed in Section 3.2.1.

3.1.3 Bus Phase Angles GMRF

We now apply the preceding to the bus phase angles. The DC power flow model [Abur and Exposito, 2004] is often used for analysis of power systems in normal operations. When the system is stable, the phase angle differences are small, so $\sin(\theta_i - \theta_j) \sim \theta_i - \theta_j$. By the DC power flow model, the system state X can be described using bus phase angles. The active power flow on the transmission line connecting bus i to bus j is given by

$$P_{ij} = b_{ij}(X_i - X_j), \tag{3.1.1}$$

where X_i and X_j denote the phasor angles at bus i and j respectively, and b_{ij} denotes the inverse of the line inductive reactance. The power injected at bus i equals the algebraic sum of the powers flowing away from bus i:

$$P_i = \sum_{j \neq i} P_{ij} = \sum_{j \neq i} b_{ij}(X_i - X_j). \tag{3.1.2}$$

When buses i and j are not connected, $b_{ij} = 0$. Thus, it follows that the phasor angle at bus i could be represented as

$$X_i = \sum_{j \neq i} \left\{ \frac{b_{ij}}{\sum_{i \neq j} b_{ij}} \right\} X_j + \frac{1}{\sum_{j \neq i} b_{ij}} P_i. \tag{3.1.3}$$

Eq. (3.1.1) can also be rewritten in matrix form as

$$P = BX, \tag{3.1.4}$$

where $P = [P_1, P_2, ..., P_p]^\top$ is the vector of injected active powers, $X = [X_1, X_2, ..., X_p]^\top$ is the vector of bus phase angles and

$$B = \begin{cases} -b_{ij} & \text{if} \quad i \neq j, \\ \sum_{j \neq i} b_{ij} & \text{if} \quad i = j. \end{cases} \tag{3.1.5}$$

Remark: Note that, because of linearity of the DC power flow model, the above equations are valid for both the phase angle X together with the injected power

P and for the *fluctuations* of the phase angle X together with the *fluctuations* of the injected power P around its steady-state value. Specifically, if we let $\widetilde{P}$ refer to the vector of active power fluctuations and $\widetilde{X}$ represent the vector of phase angle fluctuations, we have $\widetilde{P} = B\widetilde{X}$. In the following, the focus is on the DC power flow model. Nevertheless, our analysis remains valid if we consider *fluctuations* around the steady-state values.

Because of load uncertainty, and under generation-load balance, the injected power can be modeled as a random variable [Zhang and Lee, 2004]. The injected power is the sum of many random factors such as load fluctuations, wind turbine and Photo Voltaic Cell (PVC) output fluctuations, etc. While the independence of the constituting random variables can be justified, their identical distribution cannot. Therefore, using the *Lyapunov* Central Limit Theorem(CLT) [B. De Finetti, 1975, Sec. 7.7.2], which does not require the random variables to be identically distributed, we can model the injected power as a Gaussian distribution.

Lyapunov CLT: Let $\{Y_i : i = 1, 2, \ldots, n\}$ be a sequence of independent random variables each with finite expected value μ_i and variance σ_i^2. Define $s_n^2 = \sum_{i=1}^n \sigma_i^2$. If the Lyapunov condition[1] is satisfied, then $\sum_{i=1}^n \frac{(Y_i - \mu_i)}{s_n}$ converges in distribution to a standard normal random variable as n goes to infinity.

Considering conventional assumptions in power systems, the Lyapunov condition is met. As argued in [Sedghi and Jonckheere, 2014], the Gaussian assumption

[1] The condition requires that $\exists \delta > 0$ such that the random variables $|Y_i - \mu_i|$ have moments of order $2 + \delta$ and the rate of growth of these moments is limited in the sense that $\lim_{n\to\infty} \frac{\sum_{i=1}^n E|Y_i - \mu_i|^{2+\delta}}{s_n^{2+\delta}} = 0$.

is justified in the transmission network. The Gaussian model is also utilized in various analysis of power networks such as [Kashyap and Callaway, 2010, Schellenberg et al., 2005, Pang et al., 2012, Dopazo et al., 1975] where n is estimated to be of order 1000. To exemplify CLT, it is suggested in [Mur-Amada and Salln-Arasanz, April 2011] that as few as 5 wind turbines would suffice to see CLT in action. Therefore, for each i, we model P_i in Eq. (3.1.2) with a Gaussian random variable. Hence the linear relationship in Eq. (3.1.4), together with the fixed phasor at the slack bus, implies that the phasor angles θ_i are Gaussian random variables [He and Zhang, June 2011].

The next step is to find out whether the X_i's satisfy the local Markov property and, in the affirmative, to discover the neighbor sets corresponding to each node. We do this by analyzing Eq. (3.1.3). If there were only the first term, we would conclude that the set of nodes electrically connected to node i satisfies the local Markov property, but the second term makes a difference. Below, we argue that an analysis of the second term of (3.1.3) shows that this term causes some second-neighbors of X_i to have a nonzero term in matrix J . In addition, for nodes that are more than two hops apart, $J_{ij} = 0$. Therefore, as opposed to the claim in [He and Zhang, June 2011], a second-neighbor relationship *does exist* in matrix J. The second neighbor property may result in additional edges in the Markov graph between the nodes that are second neighbors in the grid graph.

As stated earlier, the powers injected at different buses have Gaussian distribution. We can assume that they are independent and without loss of generality they are zero mean. Therefore, the probability distribution function for P is $f_P(P) \propto e^{-\frac{1}{2}P^\top P}$. Since $P = BX$, we have $f_X(X) \propto e^{-\frac{1}{2}X^\top B^\top BX}$. Recalling the definition of the probability distribution function for jointly Gaussian random variables in (2.3.1), we get $J = B^T B$. Let $d(i,j)$ represent the hop distance between nodes i and j in the power grid graph G. By definition of matrix B, this leads to some nonzero J_{ij} entries for $d(i,j) = 2$. In addition, we state the following:

Proposition 1. *Assume that the powers injected at the nodes are Gaussian and mutually independent. Then*

$$J_{ij} = 0, \qquad \forall\ d(i,j) > 2.$$

Proof. We argue by contradiction. Assume $J_{ij} \neq 0$ for some $d(i,j) > 2$. Since $J_{ij} = \sum_k B_{ik} B_{jk}$, it follows that $\exists\ k\ s.t.\ B_{ik} \neq 0, B_{jk} \neq 0$. By (3.1.5), $B_{ik} \neq 0$ implies $d(i,k) = 1$. From there on, the triangle inequality implies that $d(i,j) \leq d(i,k) + d(k,j) = 1 + 1 = 2$, which contradicts the assumption $d(i,j) > 2$. ■

It was shown in [Sedghi and Jonckheere, 2014] that for some graphs, the second-neighbor terms are smaller than the terms corresponding to the immediate electrical neighbors of X_i. More precisely, it was shown that for lattice-structured grids, this approximation falls under the generic fact of the tapering off of Fourier coefficients [Sedghi and Jonckheere, 2014]. Therefore, we can approximate each neighborhood with the immediate electrical neighbors. We can also proceed with the exact relationship. For simplicity, we opt for the first-neighbor analysis. We explain shortly why CMIT works with this approximation as well.

Note that our detection method relies on the graphical model of the variables. It is based on the fact that the Markov graph of bus phase angles changes under an attack. CMIT is tuned with correct data and we prove that in case of attack, the Markov graph of compromised data does not follow the Markov graph of correct data. Hence, we can tune CMIT by either the exact relationship or the approximate Markov graph. In both cases, the output in case of attack is different from the output tuned with correct data. Therefore, CMIT works for both approximate and exact neighborhoods.

3.2 Structure Learning

In the context of graphical models, model selection means finding the exact Markov graph underlying a group of random variables based on samples of those random variables. There are two main classes of methods for learning the structure of the underlying graphical model, convex methods and non-convex methods. The ℓ_1-regularized maximum likelihood estimators are the main class of convex methods [Friedman et al., 2007, Ravikumar et al., 2011, Janzamin and Anandkumar, 2012, 2014]. In these methods, the inverse covariance matrix is penalized with a convex ℓ_1-regularizer in order to encourage sparsity in the estimated Markov

Algorithm 1 $CMIT(x^n; \xi_{n,p}, \eta)$ for structure learning using samples x^n [Anandkumar et al., 2012]

```
**Initialize** $\widehat{G}_p^n = (V, \emptyset)$
For each $i, j \in V$,
if    $\min_{\substack{S \subset V \setminus \{i,j\} \\ |S| \leq \eta}} \widehat{\Sigma}(i, j|S) > \xi_{n,p}$,
 then
  add $(i, j)$ to the edge set of $\widehat{G}_p^n$.
end if
Output: $\widehat{G}_p^n$
```

graph structure. The other types of methods are the non-convex or greedy methods [Anandkumar et al., 2012]. In our work [Sedghi and Jonckheere, 2013, 2014, 2015], we use the latter methods.

3.2.1 Conditional Covariance Test

In order to learn the structure of the power grid, we utilize the Gaussian Graphical Model Selection method called *CMIT* [Anandkumar et al., 2012]. CMIT estimates the structure of the underlying graphical model given i.i.d. samples of the random variables. CMIT is shown in Algorithm 1.

In Algorithm 1, the output is an edge set corresponding to graph G given n i.i.d. samples x^n, each of which has p variables (corresponding to vertices), a threshold $\xi_{n,p}$ (that depends on both p and n) and a constant $\eta \in \mathbb{N}$, which is related to the local vertex separation property (described later). In our case, each one of the p variables represents a bus phase angle.

The sufficient condition for output of CMIT to have structural consistency with the underlying Markov graph among variables is that the graph has to satisfy local separation property and walk-summability [Anandkumar et al., 2012]. An ensemble of graphs has the (η, γ)-local separation property if for any $(i, j) \notin E(G)$, the maximum number of paths between i and j of length at most γ does not exceed η. A Gaussian model is said to be α-walk summable if $||\bar{\mathbf{R}}|| \leq \alpha < 1$, where $\bar{\mathbf{R}} = [|r_{ij}|]$ and $||.||$ denotes the spectral or 2-norm of a matrix [Anandkumar et al., 2012]. $\mathbf{R} = [r_{ij}]$ is the matrix of partial correlation coefficients; it vanishes on the diagonal entries and on the non-diagonal entries it is given by

$$\begin{aligned} r_{ij} &\triangleq \frac{\Sigma(i,j|V \setminus \{i,j\})}{\sqrt{\Sigma(i,i|V \setminus \{i,j\})\Sigma(j,j|V \setminus \{i,j\})}} \\ &= -\frac{J(i,j)}{\sqrt{J(i,i)J(j,j)}}. \end{aligned} \tag{3.2.1}$$

r_{ij}, the *partial correlation coefficient* between variables X_i and X_j for $i \neq j$, measures their conditional covariance given all other variables [Lauritzen, 1996].

Regardless of whether the exact or approximate neighborhood relationship holds, the Markov graph of the bus phase angles is an example of bounded local path graphs that satisfy the local separation property. We also checked the analyzed networks for the walk-summability condition. As shown in (3.2.1) and the definition of walk-summability, this property depends only on matrix J and thus on the topology of the grid. The walk-summability does not depend on the operating point of the grid.

It is shown in [Anandkumar et al., 2012] that, under walk-summability, the effect of faraway nodes on the covariance decays exponentially with the distance and the error in approximating the covariance by local neighboring decays exponentially with the distance. Hence by correct tuning of threshold $\xi_{n,p}$ and with enough samples, we expect the output of CMIT to follow the grid structure.

The computational complexity of CMIT is $O(p^{\eta+2})$, which is efficient for small η [Anandkumar et al., 2012]. η is the parameter associated with local separation property described above. The sample complexity associated with CMIT is $n = \Omega(J_{\min}^{-2} \log p)$, where $J_{\min}$ is the minimum absolute edge potential in the model [Anandkumar et al., 2012].

It is worth mentioning that since we use CMIT for structure learning of phasor data, our method is robust against measurement noise. The reason is that CMIT analyzes conditional covariance of its input data. Since input data is Gaussian, the conditional covariance can be found from covariance matrix for phasor data, i.e. $\Sigma(X, X)$ (see Eq. 3.2.2). Let N be the sum of the measurement noise and systematic errors. Both systematic errors and measurement noise are independent of the measured values. Also, we know that $\mathbb{E}(X) = 0$. Therefore, $\Sigma(X + N, X + N) = \Sigma(X, X) + \Sigma(N, N)$. Note that in CMIT we only look at pairs (i, j) such that $i \neq j$. Therefore as long as $\Sigma(N, N)$ has a diagonal form, this error does not influence our performance. This is the case when errors at different locations in the network are independent of each other. Measurement noise meets this criterion. Moreover, if systematic error in the network has a diagonal covariance matrix $\Sigma(N, N)$, it also does not impact our method. Even if systematic errors do not have a diagonal covariance but remain the same with time, they can be detected and compensated during an initial training phase when we are sure the system is not under the attack.

CMIT distributes the edges fairly uniformly across the nodes, while the ℓ_1 method tends to cluster all the edges together among the “dominant” variables leading to a densely connected component and several isolated points [Anandkumar et al., 2012] and thus a disconnected graph. Therefore, the ℓ_1 method has some limitations in detecting the structure of a connected graph. The power grid transmission network is a connected graph where the edges are distributed over the network. Therefore, CMIT is more suitable for detecting the structure of the power grid.

3.2.2 Decentralization

We want to find the Markov graph of our bus phasor measurements. The connection between electrical connectivity and correlation (Proposition 1) helps us to decentralize our method to a great extent. The power network in its normal operating condition consists of different areas connected together via border nodes. A border node is any node that is also connected to a node from a different area as depicted in [bor, 2014]. Therefore, we decompose our network into these sub-areas. Our method can be performed locally in the sub-networks. The sub-network connection graph is available online from the protection system at each sub-network and can be readily compared with the bus phase angle Markov graph. In addition, only for border nodes we need to consider their out-of-area neighbors as well. This can be done either by solving the power flow equations for that border link or by receiving measurements from neighbor sub-networks. Therefore, we run CMIT for each sub-graph to figure out its Markov graph. Then we compare it with online network graph information to detect false data injection attacks.

This decentralization reduces complexity and increases speed. Our decentralized method is a substitute for considering all measurements throughout the power grid, which requires a huge amount of data exchange, computation, and overhead. In addition to having fewer nodes to analyze, this decentralization leads us to a smaller η and greatly reduces computational complexity, which makes our method capable of being executed in very large networks. Furthermore, since structure learning is performed locally, faraway relationships created by nonlinearities—ignored in Prop. 1 but intrinsic to power systems—are mitigated, hence our neighborhood assumptions are justified. Last but not least, utility companies are not willing to expose their information for economical competition reasons and there have been several attempts to make them do that [Rajagopalan et al., 2011]. Thus it is desired to reduce the amount of data exchange between different areas and our method adequately fulfills this preference.

It should be noted that the measurement vector X analyzed in our work is a mixture of measurements from PMUs and State Estimator output corresponding to the same time. This is achieved as follows. PMUs use GPS-sync time stamp and State Estimator measurements in SCADA are labeled with local time stamp. Since our method is performed locally, it has two advantages. First, as discussed earlier, it avoids large delays in communication network. Second, we can use the local time stamps from State Estimator outputs. We do not require the high rate of measurement from PMUs for our detection scheme and only consider the PMU samples at the time we have State Estimator samples. Since both data have time stamps, we are able to form the measurement vector X with measurement data from the same time.

3.2.3 Online Calculations

For fast monitoring of the power grid, we need an on-line algorithm. As we show in this section, our algorithm can be developed as an iterative method that processes new data without the need for reprocessing earlier data. Here, we derive an iterative formulation for the sample covariance matrix. Then we use it to calculate the conditional covariance using

$$\widehat{\Sigma}(i,j|S) := \widehat{\Sigma}(i,j) - \widehat{\Sigma}(i,S)\widehat{\Sigma}^{-1}(S,S)\widehat{\Sigma}(S,j). \tag{3.2.2}$$

As we know, in general,

$$\Sigma = E[(X-\mu)(X-\mu)^\top] = E[XX^\top] - \mu\mu^\top.$$

Let $\widehat{\Sigma}^{(n)}(X)$ denote the sample covariance matrix for a vector X of p elements from n samples and let $\widehat{\mu}^{(n)}(X)$ be the corresponding sample mean. In addition, let $X^{(i)}$ be the ith sample of our vector. Then we have

$$\widehat{\Sigma}^{(n)}(X) = \frac{1}{n-1}\left(\sum_{i=1}^{n} X^{(i)}{X^{(i)}}^\top\right) - \widehat{\mu}^{(n)}{\widehat{\mu}^{(n)}}^\top. \tag{3.2.3}$$

Therefore,

$$\begin{aligned}\widehat{\Sigma}^{(n+1)}(X) =& \frac{1}{n}\left[\sum_{i=1}^{n} X^{(i)}{X^{(i)}}^\top + X^{(n+1)}{X^{(n+1)}}^\top\right] && (3.2.4a)\\ &- \widehat{\mu}^{(n+1)}{\widehat{\mu}^{(n+1)}}^\top, \\ \widehat{\mu}^{(n+1)} =& \frac{1}{n+1}[n\widehat{\mu}^{(n)} + X^{(n+1)}]. && (3.2.4b)\end{aligned}$$

By keeping the first term in (3.2.3) and the sample mean (3.2.4b), our updating rule is (3.2.4a). Thus, we revise the sample covariance as soon as any bus phasor measurement changes and leverage it to reach the conditional covariances needed for CMIT. It goes without saying that if the system demand and structure does not change and the system is not subject to false data injection attack, the voltage angles at nodes remain the same and there is no need to run any algorithm.

3.3 Stealthy Deception Attack

The most recent and most dreaded false data injection attack on the power grid was introduced in [Teixeira et al., 2011]. It assumes knowledge of the bus-branch model and it is capable of deceiving the State Estimator. For a p-bus electric power network, the $l = 2p - 1$ dimensional state vector x is $[\theta^\top, V^\top]^\top$, where $V = [V_1, ..., V_p]^\top$ is the vector of voltage bus magnitudes and $\theta = [\theta_2, ..., \theta_p]^\top$ the vector of phase angles. It is assumed that the nonlinear measurement model for the state estimation is $z = h(x) + \epsilon$, where $h(.)$ is the measurement function, $z = [z_P^\top, z_Q^\top]^\top$ is the measurement vector consisting of active and reactive power flow measurements and ϵ is the measurement error. $H(x^k) := \frac{dh(x)}{dx}|_{x=x^k}$ denotes the Jacobian matrix of the measurement model $h(x)$ at x^k. The goal of the stealthy deception attacker is to compromise the measurements available to the State Estimator (SE) as $z^a = z + a$, where z^a is the corrupted measurement vector and a is the attack vector. The vector a is designed such that the SE algorithm converges and the attack a is undetected by the Bad Data Detection scheme. Then it is shown that, under the DC power flow model, such an attack can only be performed locally with $a \in \mathrm{Im}(H)$, where $H = H_{P\theta}$ is the matrix connecting the vector of bus injected active powers to the vector of bus phase angles, i.e., $P = H_{P\theta}\theta$. The attack is shown in Figure 3.2.

3.4 Stealthy Deception Attack Detection

In this Section, we show that our method can detect the aforementioned stealthy deception attack despite the fact that it remains undetected by the traditional Bad Data Detection scheme. The fundamental idea behind our detection scheme is that of *structure learning*. Our learner, CMIT, is first tuned with correct data, which corresponds to the grid graph. Therefore, any attack that changes the structure alters the output of CMIT and this triggers the alarm. Let us consider the attack more specifically. As we are considering the DC power flow model and all voltage magnitudes are normalized to 1 p.u., the state vector introduced in [Teixeira et al.,

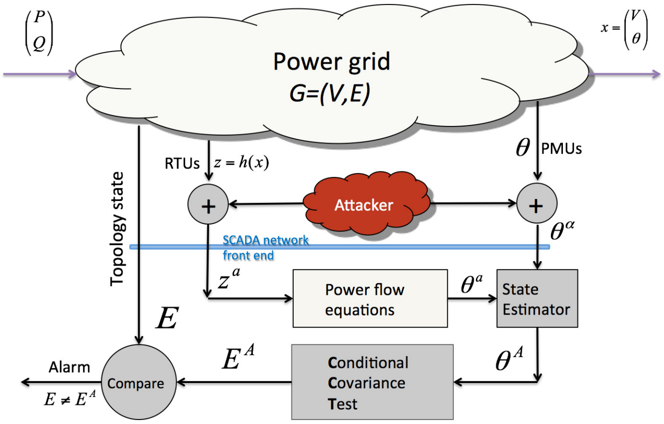


Figure 3.2: Power grid under a cyber attack

2011] reduces to the vector of voltage angles, X. Since $a \in \mathrm{Im}(H)$, $\exists\ d$ such that $a = Hd$ and

$$z^a = z + a = H(X + d) = HX^a,$$

where X^a represents the vector of angles when the system is under attack, z^a is the attacked measurement vector, and X is the correct phasor angle vector. Considering (3.1.2), we have $H_{ij} = -b_{ij}$ for $i \neq j$ and $H_{ii} = \sum_{i \neq j} b_{ij}$, where b_{ij} denotes the inverse of the line inductive reactance. We have

$$X^a = X + d = H^{-1}P + H^{-1}a = H^{-1}(P + a). \tag{3.4.1}$$

As the definition of matrix H shows, it is of rank $p - 1$. Therefore, the above H^{-1} denotes the pseudo-inverse of matrix H. Another way to address this singularity is to remove the row and the column associated with the slack bus. From (3.4.1), we get

$$\begin{aligned}\Sigma(X^a, X^a) &= H^{-1}[\Sigma(P + a, P + a)]H^{-1^T} \\ &= H^{-1}[\Sigma(P, P) + \Sigma(a, a)]H^{-1^T}.\end{aligned}$$

The above calculation assumes that the attack vector is independent of the current measurement values in the network, as demonstrated in the definition of the attack [Teixeira et al., 2011].

An attack is considered successful if it causes the operator to make a wrong decision. For that matter, the attacker would not insert just one wrong sample. In addition, if the attack vector remains constant, it does not cause any reaction. This eliminates the case of constant attack vectors. Therefore, the attacker is expected to insert non-constant vectors a during some samples. Thus $\Sigma(a, a) \neq 0$ and

$$\Sigma(X^a, X^a) \neq \Sigma(X, X). \tag{3.4.2}$$

It is not difficult to show that, if we remove the assumption on independence of attack vector and the injected power, (3.4.2) still holds.

Considering (3.4.2) and the fact that matrix inverse is unique, it follows that, in case of an attack, the new Σ^{-1} will not be the same as the network information matrix in normal condition, i.e., $\Sigma^{-1}(X^a, X^a) \neq J_{\text{normal}}$, and as a result, the output of CMIT will not follow the grid structure. We use this mismatch to trigger the alarm. It should be noted that acceptable load changes do not change the Markov graph and as a result do not lead to false alarms. The reason is that such changes do not falsify the DC power flow model and the Markov graph will continue to follow the defined information matrix. After the alarm is triggered, the next step is to find which nodes are under attack.

3.4.1 Detecting the Set of Attacked Nodes

We use the *correlation anomaly* metric [Ide T. Lozano, 2009] to find the attacked nodes. This metric quantifies the contribution of each random variable to the difference between two probability densities while considering the sparsity of the structure. The Kullback-Leibler (KL) divergence is used as the measure of the difference. As soon as an attack is detected, we use the attacked information matrix and the information matrix corresponding to the current topology of the grid to compute the anomaly score for each node. The nodes with highest anomaly scores are announced as the nodes under attack. We investigate the implementation details in the next section.

It should be noted that the attack is performed locally and because of the local Markov property, we are certain that no nodes from other sub-graphs contribute to the attack.

We should emphasize that the considered attack assumes the knowledge of the system bus-branch model. Therefore, the attacker is equipped with very critical information. Yet, we can mitigate such an "intelligent" attack.

3.4.2 Reactive Power versus Voltage Amplitude

As mentioned before, with similar calculations, we can consider the case where the attacker manipulates reactive power data to lead the State Estimator to wrong estimates of the voltage. Such an attack can be designed to fake a voltage collapse or trick the operator to cause a change in the normal state of the grid. For example, if the attacker fakes a decreasing trend in the voltage magnitude in some part of the grid, the operator will send more reactive power to that part and thus this could cause voltage overload/underload. At this point, the protection system would disconnect the corresponding lines. This could lead to outages in some areas and in a worse scenario to overloading in other parts of the grid that might cause blackouts and cascading events.

The detection can be done by linearization of the AC power flow and by considering the fluctuations around steady state. Then pursuing our algorithm, it readily follows that such an attack can also be detected with a similar approach to the one developed here for bus phase angles and active power.

In the rest of this section, we show how this analogy can be established. The AC power flow states that the active power and the reactive power flowing from bus i to bus j are, respectively,

$$P_{ij} = G_{ij}V_i^2 - G_{ij}V_iV_j\cos(\theta_i - \theta_j) + b_{ij}V_iV_j\sin(\theta_i - \theta_j),$$

$$Q_{ij} = b_{ij}V_i^2 - b_{ij}V_iV_j\cos(\theta_i - \theta_j) - G_{ij}V_iV_j\sin(\theta_i - \theta_j),$$

where V_i and θ_i are the voltage magnitude and phase angle, resp., at bus i and G_{ij} and b_{ij} are the conductance and susceptance, resp., of line ij. From [Banirazi and Jonckheere, 2010], we obtain the following approximation of the AC *fluctuating* power flow:

$$\begin{aligned}
&\widetilde{P}_{ij} = (b_{ij}\overline{V}_i\overline{V}_j\cos\overline{\theta}_{ij})(\widetilde{\theta}_i - \widetilde{\theta}_i),\\
&\widetilde{Q}_{ij} = (2b_{ij}\overline{V}_i - b_{ij}\overline{V}_j\cos\overline{\theta}_{ij})\tilde{V}_i - (b_{ij}\overline{V}_i\cos\overline{\theta}_{ij})\widetilde{V}_j,
\end{aligned}$$

where an overbar denotes the steady-state value, a tilde means the fluctuation around the steady-state value, and $\overline{\theta}_{ij} = \overline{\theta}_i - \overline{\theta}_j$. These fluctuating values due to renewables and variable loads justify the utilization of probabilistic methods in power grid problems.

Now assuming that for the steady-state values of the voltages we have $\overline{V}_i = \overline{V}_j \simeq 1$ $p.u.$ (per unit) and the fluctuations in angles are about the same such that $\cos\theta_{ij} = 1$, we have

$$\widetilde{P}_{ij} = b_{ij}(\widetilde{\theta}_i - \widetilde{\theta}_j), \tag{3.4.3a}$$

$$\widetilde{Q}_{ij} = b_{ij}(\widetilde{V}_i - \widetilde{V}_j). \tag{3.4.3b}$$

It is clear from (3.4.3a)-(3.4.3b) that we can follow the same approach we had about active power and voltage angles with reactive power and voltage magnitudes, respectively.

It can be argued that, as a result of uncertainty, the aggregate reactive power at each bus can be approximated as a Gaussian random variable and, because of

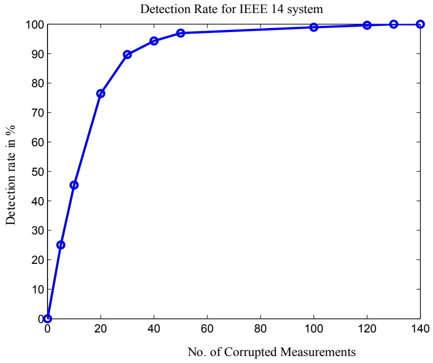


Figure 3.3: Detection rate for IEEE-14 bus system

Eq. $(3.4.3b)$, the voltage fluctuations around the steady-state value can be approximated with Gaussian random variables. Therefore, the same path of approach as for phase angles can be followed to show the GMRF property for voltage amplitudes. Comparing $(3.4.3b)$ with (3.1.1) makes it clear that the same matrix, i.e., matrix B developed in Section 3.1.3, is playing the role of correlating the voltage amplitudes. Therefore, assuming that the statistics of the active and reactive power fluctuations are similar, the underlying graph is the same. This can readily be seen by comparing $(3.4.3a)$ and $(3.4.3b)$.

3.5 Simulation

Training the System We consider IEEE-14 bus system as well as IEEE-30 bus system. First, we feed the system with Gaussian demand and simulate the power grid. We use MATPOWER [Zimmerman et al., Feb. 2011] for solving the DC power flow equations for various demand and use the resulting angle measurements as the input to CMIT. We leverage YALMIP [Lofberg, 2004] and SDPT3 [Toh et al., 1999] to run CMIT in MATLAB. With the right choice of parameters and threshold $\xi_{n,p}$ of CMIT, and enough measurements, the Markov graph should follow the grid structure. We use the edit distance between two graphs for tuning the threshold $\xi_{n,p}$. The edit distance between two graphs reveals the number of edges that exist in only one of the two graphs.

Detecting Attack State After the threshold $\xi_{n,p}$ is set, our detection algorithm works in the following manner. Each time the procedure is initiated, i.e., when any PMU angle measurement or State Estimator output changes, it updates the conditional covariances $\hat{\Sigma}(i,j|S)$ based on new data, runs CMIT and checks the edit distance between the Markov graph of phasor data and the grid structure. A discrepancy triggers the alarm. Subsequently to an alarm, the system uses anomaly metric to find all the buses under the attack. The flowchart of our method is shown in Figure 3.1.

Next, we introduce the stealthy deception attack on the system. The attack is designed according to the description in [Teixeira et al., 2011], i.e., it is a random vector such that $a \in \text{Im}(H)$. The attack is claimed to be successful only if performed locally on connected nodes. Having this constraint in mind, for IEEE-14 test case the maximum number of attacked nodes is 6 and for IEEE-30 bus system this number is 8. For the IEEE-14 network, we consider the cases where 2 to 6 nodes are under attack. For the IEEE-30 network, we consider the cases where 2 to 8 nodes are under attack. For each case and for each network, we simulate all possible attack combinations. This is to make sure we have checked our detection scheme against all possible stealthy deception attacks. Each case is repeated 1000 times for different attack vector values.

When the attacker starts tampering with the data, the corrupted samples are added to the sample bin of CMIT and are therefore used in calculating the sample covariance matrix. With enough corrupted samples, our algorithm can get arbitrarily close to 100% successful in detecting all cases of attacks discussed above, for both IEEE-14 and IEEE-30 bus systems. This is shown in Figure 3.3 for IEEE-14 bus system. The detection rate is averaged over all possible attack scenarios. The reason behind the trend shown in Figure 3.3 is that first, for a very small number of corrupted measurements, the Markov graph follows the true information matrix and then, for a higher number of compromised measurements, the Markov graph follows the random relationship that the attacker is producing. When the number of compromised samples increases, they gain more weight in the sample covariance, and the chance of a change in the Markov graph increases. It can be seen that even

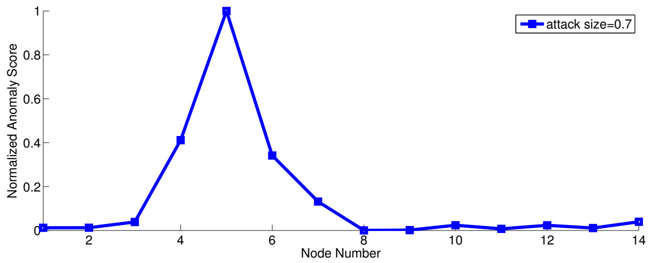


Figure 3.4: Anomaly score for IEEE-14 bus system. Nodes 4, 5, 6 are under attack; Attack size is 0.7.

for a small number of corrupted measurements, our method presents a good performance: the detection rate is 90% with 30 corrupted samples. The minimum number of corrupted samples to get almost 100% detection rate for IEEE-14 bus system is 130 and it is 50 for IEEE-30 bus system. Since IEEE-30 is more sparse than IEEE-14 bus system, our method performs more efficiently in the former case. Yet, for a 60 Hz system, the detection speed for IEEE-14 bus system is quite amazing as well.

Identifying Nodes under Attack The next step is to find which nodes are under attack. As stated earlier, we use anomaly score metric [Ide T. Lozano, 2009] to detect such nodes. As an example, Figure 3.4 shows the anomaly score plot for the case where nodes $4, 5$ and 6 are under attack[2]. It means that a random vector is added to the measurements at these nodes. This attack is repeated 1000 times for different values building an attack size of 0.7. The attack size refers to the expected value of the Euclidean norm of the attack vector a.

[2] The numbering system employed here is the one of the published IEEE-14 system available at https://www.ee.washington.edu/research/pstca/pf14/pg_tca14bus.htm

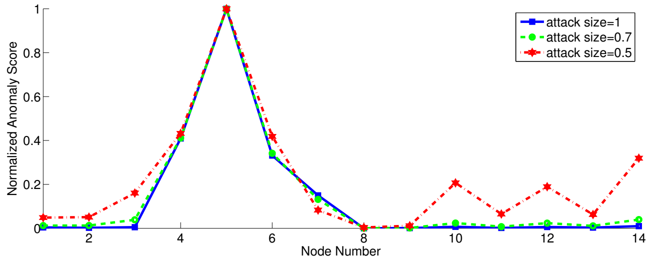


Figure 3.5: Anomaly score for IEEE-14 bus system for different attack sizes. Nodes 4, 5, 6 are under attack. Attack sizes are 0.5, 0.7, 1.

Simulation results show that as the attack size increases, the difference between the anomaly scores of the nodes under the attack and the uncompromised nodes increases and, as a result, it becomes easier to pinpoint the attacked nodes. For example, Figure 3.5 compares the cases where the attack size is 1, 0.7 and 0.5 for the attack scenario where nodes 4, 5, 6 are under attack. It should be noted that in order for an attack to be successful in misleading the TSO, the attack size should not be too small. More specifically, the attacker wants to make a change in the system state such that the change is noticeable with the hope that this would result in the wrong reaction of the TSO. If the value of the system state under the attack is close to its real value, the system is not considered under the attack as it continues its normal operation. It can be seen that, even for the smallest possible attack size that would normally not lead the operator to react, the anomaly score plot will remain reliable. For example, in the considered attack scenario, the anomaly plot performs well even for an attack size of 0.3, while it seems that a potentially successful attack under normal standards needs a bigger attack size.

Setting up Anomaly Score Threshold Setting the threshold for anomaly score is another important aspect of the detection algorithm. As discussed earlier, our scheme has two major parts. First, detection of attack state, i.e. to declare if the system is under attack. Second, the identification of the attacked nodes in case of an attack state. In Figure 3, we analyzed the detection rate of the "attack state" versus the number of corrupted samples. In Figure 4 and 5, we discussed how normalized anomaly score changes with different attack sizes. Now, we use this intuition to design the threshold for anomaly score. In case of "attack state" we calculate the normalized anomaly score for each node. For any node, if this benchmark is greater than the threshold, the node is considered to be under attack. In this context, we define the "Node Detection Ratio (NDRo)" as the ratio of the number of attacked nodes that are correctly labeled as attacked to the total number of attacked nodes. Consequently, the "False Alarm Ratio (FARo)," not to be confused with the False Alarm rate (FAR), refers to the number of uncompromised nodes that are mislabeled as under attack to the total number of uncompromised nodes. As in detection theory, there is a trade-off in designing this threshold value. Lower threshold values result in higher NDRo and higher FARo and vice versa. Since our goal is to detect all attacked nodes, we design the threshold such that the NDRo is approximately 100% with a very low FARo. To design the threshold, we repeat the simulation discussed for Figures 4 and 5 for five different sets of attacked nodes, the three discussed attack sizes, and repeat each attack size 100 times. As can also be seen in the above plots, with a threshold of 0.3 for all attacked nodes, the normalized anomaly score is above the threshold. Next, we use this threshold in all possible sets of attacked nodes on IEEE-14 bus system with a attack size of 0.7 and repeat it 50 times for each set. Simulation results show that this threshold guarantees nearly 100% NDRo with a very low FARo of 3.82×10^{-5}. The reason is that anomaly score provides a precise statistical analysis of the nodes that contribute to the mismatch. Hence, we can obtain 100% detection rate with a very low FARo.

3.6 Discussion and Conclusion

We have proposed a decentralized false data injection attack detection scheme that is capable of detecting the most recent stealthy deception attack on power grid. To the best of our knowledge, our remedy is the first to comprehensively detect this sophisticated attack. In addition to detecting the attack state, our algorithm is capable of pinpointing the set of attacked nodes. Although [Giani et al., January 2012] considers the same attack on the power network, considerable progress is made in our approach versus the one in [Giani et al., January 2012]. In both cases, the goal is to detect the attack. While [Giani et al., January 2012] seeks a PMU placement method, our method does not require additional hardware but rather performs statistical structure learning on the measurement data. In general, both PMU placement and structure learning are NP hard. However, the use of common knowledge of the grid structure helps us reach a polynomial time solution. The power network structure is a sparse graph that satisfies the local separation property and the walk-summability. For details on how these properties reduce the general NP hard problem to a tractable polynomial time problem, see [Anandkumar et al., 2012].

As stated earlier, the computational complexity of our method is polynomial and the decentralized property makes our scheme suitable for huge networks, yet with bearable complexity and run time. In addition, our method is capable of detecting attacks that manipulate reactive power measurements to cause inaccurate voltage amplitude data. Such attack scenario can lead to, or mimic a voltage collapse.

In conclusion, we have introduced change detection for the graphical model of a power system and showed that it can be used to detect data manipulation. Our method protects the power system against a large class of false data injection attacks, which is of paramount importance for current and future grid reliability, security, and stability.

Chapter 4

Stochastic Optimization in High Dimension

4.1 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 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 employ it in our work.

In our work [Sedghi et al., 2014b,a], we design 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].

We consider a simple modification to the (inexact) stochastic ADMM method [Ouyang et al., 2013] by incorporating multiple steps or epochs, which can be viewed as a form of annealing. We establish that this simple modification has huge implications in achieving tight convergence rates as the dimensions of the problem instances scale. In each iteration of the method, we employ projections on to certain norm balls of appropriate radii, and we decrease the radii in epochs over time. The idea of annealing was first introduced by Agarwal et al. [2012b] for dual averaging. Yet, that method cannot be extended for multivariable cases.

For instance, for the sparse optimization problem, we constrain the optimal solution at each step to be within an ℓ_1-norm ball of the initial estimate, obtained at the beginning of each epoch. At the end of the epoch, an average is computed and passed on to the next epoch as its initial estimate. Note that the ℓ_1 projection can be solved efficiently in linear time, and can also be parallelized easily [Duchi et al., 2008].

For matrix decomposition with a general loss function, the ADMM method requires multiple blocks for updating the low rank and sparse components. We apply the same principle and project the sparse and low rank estimates on to ℓ_1 and nuclear norm balls, and these projections can be computed efficiently.

Theoretical implications: The above simple modifications to ADMM have huge implications for high-dimensional problems. For sparse optimization, our convergence rate is $\mathcal{O}(\frac{s \log d}{T})$, for s-sparse problems in d dimensions in T steps. Our bound has the best of both worlds: efficient high-dimensional scaling (as $\log d$) and efficient convergence rate (as $\frac{1}{T}$). This also matches the minimax lower bound for the linear model and square loss function [Raskutti et al., 2011], which implies that our guarantee is unimprovable by any (batch or online) algorithm (up to constant factors). For matrix decomposition, our convergence rate is $\mathcal{O}((s + r)\beta^2(p)\log p/T)) + \mathcal{O}(\max\{s + r, p\}/p^2)$ for a $p \times p$ input matrix in T steps, where the sparse part has s non-zero entries and low rank part has rank r. For many natural noise models (e.g. independent noise, linear Bayesian networks), $\beta^2(p) = p$, and the resulting convergence rate is minimax-optimal. Note that our bound is not only on the reconstruction error, but also on the error in recovering the sparse and low rank components. These are the first convergence guarantees for online matrix decomposition in high dimensions. Moreover, our convergence rate holds *with high probability* when noisy samples are input, in contrast to expected convergence rate, typically analyzed in literature. See Table 4.1, 4.2 for comparison of this work with related frameworks.

Practical implications: The proposed algorithms provide significantly faster convergence in high dimension and better robustness to noise. For sparse optimization, our method has significantly better accuracy compared to the stochastic ADMM method and better performance than RADAR, based on multi-step dual averaging [Agarwal et al., 2012b]. For matrix decomposition, we compare our method with the state-of-art inexact ALM [Lin et al., 2010] method. While both methods have similar reconstruction performance, our method has significantly better accuracy in recovering the sparse and low rank components.

Related Work: ADMM: Existing online ADMM-based methods lack high-dimensional guarantees. They scale poorly with the data dimension (as $\mathcal{O}(d^2)$), and also have slow convergence for general problems (as $\mathcal{O}(\frac{1}{\sqrt{T}})$). Under strong convexity, the convergence rate can be improved to $\mathcal{O}(\frac{1}{T})$ but only in *expectation*: such analyses ignore the per sample error and consider only the expected convergence rate (see Table 4.1). In contrast, our bounds hold with high probability. Some stochastic ADMM methods, Goldstein et al. [2012], Deng [2012] and Luo [2012], provide faster rates for stochastic ADMM, than the rate noted in Table 4.1. 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 to be strongly convex.

It is also worth mentioning that our method provides error contraction, i.e., we can show error shrinkage after specific number of iterations whereas no other ADMM based method can guarantee this.

Related Work: Sparse Optimization: For the sparse optimization problem, ℓ_1 regularization is employed and the underlying true parameter is assumed to be sparse. This is a well-studied problem in a number of works (for details, refer to [Agarwal et al., 2012b]). Agarwal et al. [2012b] propose an efficient online method based on annealing dual averaging, which achieves the same optimal rates as the ones derived in our work. The main difference is that our ADMM method is capable of solving the problem for multiple random variables and multiple conditions while their method cannot incorporate these extensions.

Related Work: Matrix Decomposition: 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. Note that they only provide convergence rate on difference between loss function and optimal loss, whereas we provide the convergence rate on individual errors of the sparse and low rank components $\|\bar{S}(T)-S^*\|_{\mathbb{F}}^2, \|\bar{L}(T)-L^*\|_{\mathbb{F}}^2$. See Table 4.2 for comparison of guarantees for matrix decomposition problem.

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 convergence rate 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.

Note that when only the weak diffusivity condition is assumed, the matrix decomposition problem suffers from an approximation error, i.e. an error even in the noiseless setting. Both the minimax rate and the batch rates 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.

Overview of Proof Techniques: Note that in the main text, we provide guarantees for fixed-epoch length. However, if we use variable-length epoch size we can get a $\log d$ improvement in the convergence rate. Our proof involves the following high-level steps to establish the convergence rate: (1) deriving convergence rate for the modified ADMM method (with variable-length epoch size) 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. We can show that with the proposed parameter setting and varying epoch size, error can be halved by the end of each epoch. 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 match the batch guarantees for many interesting statistical models. (3) Next, we analyze how guarantees change

Method	Assumptions	convergence
ST-ADMM [Ouyang et al., 2013]	L, convexity	$\mathcal{O}(d^2/\sqrt{T})$
ST-ADMM [Ouyang et al., 2013]	SC, E	$\mathcal{O}(d^2 \log T/T)$
BADMM [Wang and Banerjee, 2013]	convexity, E	$\mathcal{O}(d^2/\sqrt{T})$
RADAR [Agarwal et al., 2012b]	LSC, LL	$\mathcal{O}(s \log d/T)$
REASON 1 (this work)	LSC, LL	$\mathcal{O}(s \log d/T)$
Minimax bound [Raskutti et al., 2011]	Eigenvalue conditions	$\mathcal{O}(s \log d/T)$

Table 4.1: *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, E = in Expectation
The last row provides minimax-optimal rate on error for any method. The results hold with high probability unless otherwise mentioned.

for fixed epoch length. We prove that although the error halving stops after some iterations but the error does not increase noticeably to invalidate the analysis.

4.2 Problem Formulation

Consider the optimization problem

$$\theta^* \in \arg\min_{\theta \in \Omega} \mathbb{E}[f(\theta, x)], \tag{4.2.1}$$

where $x \in \mathbb{X}$ is a random variable and $f : \Omega \times \mathbb{X} \to \mathbb{R}$ is a given loss function. Since only samples are available, we employ the empirical estimate of $\widehat{f}(\theta) :=$

Method	Assumptions	Convergence rate
Multi-block-ADMM [Wang et al., 2013b]	L, SC, E	$\mathcal{O}(\frac{p^4}{T})$
Batch method [Agarwal et al., 2012a]	LL, LSC, DF	$\mathcal{O}(\frac{s\log p+rp}{T})+\mathcal{O}(\frac{s}{p^2})$
REASON 2 (this work)	LSC, LL, DF	$\mathcal{O}(\frac{(s+r)\beta^2(p)\log p}{T})+\mathcal{O}(\frac{\max\{s+r,p\}}{p^2})$
Minimax bound [Agarwal et al., 2012a]	ℓ_2, IN, DF	$\mathcal{O}(\frac{s\log p+rp}{T})+\mathcal{O}(\frac{s}{p^2})$

Table 4.2: *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, IN = Independent noise model, 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-optimal rate on error for any method under the independent noise model. The results hold with high probability unless otherwise mentioned.
For Multi-block-ADMM [Wang et al., 2013b] the convergence rate is on the difference of loss function from optimal loss, for the rest of works in the table, the convergence rate is on $\|\bar{S}(T) - S^*\|_{\mathbb{F}}^2 + \|\bar{L}(T) - L^*\|_{\mathbb{F}}^2$.

$1/n \sum_{i\in[n]} f(\theta, x_i)$ in the optimization. For high-dimensional θ, we need to impose a regularization $\mathcal{R}(\cdot)$, and

$$\widehat{\theta} := \arg\min\{\widehat{f}(\theta) + \lambda_n \mathcal{R}(\theta)\}, \tag{4.2.2}$$

is the batch optimal solution.

For concreteness we focus on the 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

$$\widehat{\theta} = \arg\min \{\widehat{f}(\theta) + \lambda_n \|\theta\|_1\} \tag{4.2.3}$$

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

$$\begin{aligned} \widehat{M} &:= \arg\min\{f(M) + \lambda_n\|S\|_1 + \mu_n\|L\|_*\} \\ s.t. \quad M &= S + L, \quad \|L\|_\infty \leq \frac{\alpha}{p}. \end{aligned} \tag{4.2.4}$$

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 (4.2.2) 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 (4.2.3), where the ADMM has double blocks,and then extend it to the sparse+low rank setting of (4.2.4), which involves multi-block ADMM.

4.3 ℓ_1 Regularized Stochastic Optimization

We consider the optimization problem $\theta^* \in \arg\min \mathbb{E}[f(\theta, x)]$, $\theta \in \Omega$ where θ^* is a sparse vector. The loss function $f(\theta, x_k)$ is a function of a parameter $\theta \in \mathbb{R}^d$ and samples x_i. In stochastic setting, we do not have access to $\mathbb{E}[f(\theta, x)]$ nor to its subgradients. In each iteration we have access to one noisy sample. In order to impose sparsity we use regularization. Thus we solve a sequence

$$\theta_k \in \underset{\theta \in \Omega'}{\arg\min}\ f(\theta, x_k) + \lambda \|\theta\|_1, \quad \Omega' \subset \Omega, \tag{4.3.1}$$

where the regularization parameter $\lambda > 0$ and the constraint sets Ω' change from epoch to epoch.

Algorithm 2 Regularized Epoch-based Admm for Stochastic Optimization in high-dimensioN 1 (REASON 1)

```
**Input** $\rho, \rho_x > 0$, epoch length $T_0$ , initial prox center $\tilde{\theta}_1$, initial radius $R_1$, regularization parameter $\{\lambda_i\}_{i=1}^{k_T}$.
**Define** $Shrink_\kappa(\cdot)$ shrinkage operator in (4.3.3)
**for** Each epoch $i = 1, 2, ..., k_T$ **do**
    Initialize $\theta_0 = y_0 = \tilde{\theta}_i$
    **for** Each iteration $k = 0, 1, ..., T_0 - 1$ **do**
        $$\theta_{k+1} = \underset{\|\theta-\tilde{\theta}_i\|_1 \leq R_i}{\arg\min} \{\langle \nabla f(\theta_k), \theta - \theta_k \rangle - \langle z_k, \theta - y_k \rangle + \frac{\rho}{2}\|\theta - y_k\|_2^2 + \frac{\rho_x}{2}\|\theta - \theta_k\|_2^2\} \tag{4.3.2}$$
        $y_{k+1} = \text{Shrink}_{\lambda_i/\rho}(\theta_{k+1} - \frac{z_k}{\rho})$
        $z_{k+1} = z_k - \tau(\theta_{k+1} - y_{k+1})$
    **end for**
    **Return** : $\overline{\theta}(T_i) := \frac{1}{T}\sum_{k=0}^{T_0-1} \theta_k$ for epoch $i$ and $\tilde{\theta}_{i+1} = \overline{\theta}(T_i)$.
    **Update** : $R_{i+1}^2 = R_i^2/2$.
**end for**
```

4.3.1 Epoch-based Online ADMM Algorithm

We now describe the modified inexact ADMM algorithm for the sparse optimization problem in (4.3.1), and refer to it as REASON 1, see Algorithm 2. We consider epochs of length T_0, and in each epoch i, 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

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

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 4.7.1.

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].

$$\text{Shrink}_\kappa(a) = (a - \kappa)_+ - (-a - \kappa)_+ \tag{4.3.3}$$

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

4.3.2 High-dimensional Guarantees

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

Assumption A1: Local strong convexity (LSC) : The function $f : S \to \mathbb{R}$ 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., 2012a,b].

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: Local Lipschitz condition: For each $R > 0$, there is a constant $G = G(R)$ such that

$$|f(\theta_1) - f(\theta_2)| \leq G\|\theta_1 - \theta_2\|_1 \tag{4.3.4}$$

for all $\theta_1, \theta_2 \in S$ such that $\|\theta - \theta^*\|_1 \leq R$ and $\|\theta_1 - \theta^*\|_1 \leq R$.

We choose the algorithm parameters as below where λ_i is the regularization for ℓ_1 term, ρ and ρ_x are penalties in θ-update as in (4.3.2) and τ is the step size for the dual update.

$$\begin{aligned} \lambda_i^2 &= \frac{\gamma}{s\sqrt{T_0}}\sqrt{R_i^2 \log d + \frac{G^2 R_i^2}{T_0} + \sigma_i^2 R_i^2 w_i^2} \\ \rho &\propto \frac{\sqrt{T_0 \log d}}{R_i}, \quad \rho_x > 0, \quad \tau = \rho. \end{aligned} \tag{4.3.5}$$

Theorem 1. *Under Assumptions $A1 - A3$, λ_i as in* (4.3.5) *, we use fixed epoch length $T_0 = T \log d / k_T$ where T is the total number of iterations. Assuming this setting ensures $T_0 = \mathcal{O}(\log d)$, for any θ^* with sparsity s, we have*

$$\|\bar{\theta}_T - \theta^*\|_2^2 = \mathcal{O}\left(s \; \frac{\log d + (w^2 + \log(k_T/\log d))\sigma^2}{T} \; \frac{\log d}{k_T}\right),$$

with probability at least $1 - 3\exp(w^2/12)$, where $\bar{\theta}_T$ is the average for the last epoch for a total of T iterations and

$$k_T = \log_2 \frac{\gamma^2 R_1^2 T}{s^2(\log d + 12\sigma^2 w^2)}.$$

For proof, see Appendix A.2.6.

Improvement of $\log d$ factor : The above theorem covers the practical case where the epoch length T_0 is fixed. We can improve the above results using varying epoch lengths (which depend on the problem parameters) such that $\|\bar{\theta}_T - \theta^*\|_2^2 = \mathcal{O}(s \log d/T)$. See Theorem 3 in Appendix A.1.

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 et al., 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 difference is our method is capable of solving problems with multiple variables and constraints, as discussed in the next section, while RADAR cannot be generalized to do so.

Remark on Lipschitz property: In fact, our method requires a weaker condition than local Lipschitz property. We only require the following bounds on the dual variable: $\|z_{k+1}-z_k\|_1$ and $\|z_k\|_\infty$. Both these are upper bounded by $G+2(\rho_x+\rho)R_i$. In addition the ℓ_1 constraint does not influence the bound on the dual variable. For details see Section A.2.1.

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 convergence rate consists of $\langle e_k, \theta_k - \hat{\theta}_i \rangle$, where e_k is the error in the gradient (see Appendix A.2.2). 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 convergence rate. 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 convergence rate in the above theorem. Moreover, this ℓ_1 constraint can be efficiently implemented, as discussed in Section 4.3.1.

4.4 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 $L \in \mathbb{R}^{p\times p}$ based on the loss function f on $M = S+L$. The batch program is given in Equation (4.2.4) and we now design an online program based on multi-block ADMM algorithm, where the updates for M, S, L are carried out independently.

In the stochastic setting, we consider the optimization problem $M^* \in \arg\min \mathbb{E}[f(M,X)]$, where we want to decompose M into a sparse matrix $S \in \mathbb{R}^{p\times p}$ and a low rank matrix $L \in \mathbb{R}^{p\times p}$. $f(M, X_k)$ is a function of parameter M and samples X_k. X_k can be a matrix (e.g. independent noise model) or a vector (e.g. Gaussian graphical model). In stochastic setting, we do not have access to $\mathbb{E}[f(M,X)]$ nor to its subgradients. In each iteration we have access to one noisy sample and update our estimate based on that. We impose the desired properties with regularization. Thus, we solve a sequence

$$\begin{aligned} M_k :=& \arg\min\{\widehat{f}(M, X_k) + \lambda\|S\|_1 + \mu\|L\|_*\} \\ & s.t. \;\; M = S + L, \;\; \|L\|_\infty \leq \frac{\alpha}{p}. \end{aligned} \tag{4.4.1}$$

4.4.1 Epoch-based Multi-Block ADMM Algorithm

We now extend the ADMM method proposed in REASON 2 to multi-block ADMM. The details are in Algorithm 3, 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

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

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 4.5.2.

$$\underset{\|S-\tilde{S}_i\|_1^2 \le R_i^2}{\arg\min} \quad \lambda_i \|S\|_1 + \frac{\rho}{2\tau_k} \|S - (S_k + \tau_k G_{M_k})\|_{\mathbb{F}}^2, \tag{4.4.2}$$

$$\underset{\substack{\|L-\tilde{L}_i\|_*^2 \le \tilde{R}_i^2 \\ \|L\|_\infty \le \alpha/p}}{\arg\min} \quad \lambda_i \|L\|_* + \frac{\rho}{2\tau_k} \|L - (L_k + \tau_k G_{M_k})\|_{\mathbb{F}}^2, \tag{4.4.3}$$

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

As before, we consider epochs of length T_0 and project 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 project for the sparse estimate S. For the low rank estimate L, we impose a nuclear norm projection around the epoch initialization $\tilde{L}_i$. Intuitively, the nuclear norm projection , which is an ℓ_1 projection 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 (4.2.4) in [Agarwal et al., 2012a], and we assume that the true matrix L^* satisfies this constraint. For more discussions, see Section 4.4.2.

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 Section 4.3.1. For implementation details see Appendix 4.7.2.

For the L-update, we introduce an additional auxiliary variable Y and we have

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

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

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

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

4.4.2 High-dimensional Guarantees

We now provide guarantees that REASON 2 efficiently recovers both the sparse and the low rank estimates in high dimensions efficiently. We need the following assumptions, 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: Local strong convexity (LSC) The function $f : \mathbb{R}^{d_1 \times d_2} \rightarrow \mathbb{R}^{n_1 \times n_2}$ satisfies an R-local form of strong convexity (LSC) if there is a non-negative constant $\gamma = \gamma(R)$ such that $f(B_1) \geq f(B_2) + \mathrm{Tr}(\nabla f(B_2)(B_1 - B_2)) + \frac{\gamma}{2}\|B_2 - B_1\|_{\mathbb{F}}$, for any $\|B_1\| \leq R$ and $\|B_2\| \leq R$, which is essentially the matrix version of

Assumption A1. Note that we only require LSC condition on $S+L$ and not jointly on S and L.

We choose algorithm parameters as below where λ_i, μ_i are the regularization for ℓ_1 and nuclear norm respectively, ρ, ρ_x correspond to penalty terms in M-update and τ is dual update step size.

$$\begin{aligned} \lambda_i^2 &= \frac{\gamma\sqrt{R_i^2+\tilde{R}_i^2}}{(s+r)\sqrt{T_0}}\sqrt{\log p + \frac{G^2}{T_0} + \beta^2(p)\sigma_i^2 w_i^2} \\ &\quad + \frac{\rho_x^2(R_i^2+\tilde{R}_i^2)}{T_0} + \frac{\alpha^2}{p^2} + \frac{\beta^2(p)\sigma^2}{T_0}\left(\log p + w_i^2\right), \\ \mu_i^2 &= c_\mu \lambda_i^2, \quad \rho \propto \sqrt{\frac{T_0 \log p}{R_i^2+\tilde{R}_i^2}}, \quad \rho_x > 0, \quad \tau = \rho. \end{aligned} \tag{4.4.4}$$

Theorem 2. *Under assumptions* $A2-A6$*, parameter settings* (4.4.4) *, let* T *denote total number of iterations and* $T_0 = T\log p/k_T$*. Assuming that above setting guarantees* $T_0 = \mathcal{O}(\log p)$*,*

$$\begin{aligned} &\|\bar{S}(T)-S^*\|_{\mathbb{F}}^2 + \|\bar{L}(T)-L^*\|_{\mathbb{F}}^2 = \\ &\mathcal{O}\left((s+r)\frac{\log p + \beta^2(p)\sigma^2\left(w^2+\log(k_T/\log p)\right)}{T}\frac{\log p}{k_T}\right) + \left(1+\frac{s+r}{\gamma^2 p}\right)\frac{\alpha^2}{p}, \end{aligned} \tag{4.4.5}$$

with probability at least $1-6\exp(-w^2/12)$*,*

$$k_T \simeq -\log\left(\frac{(s+r)^2}{\gamma^2 R_1^2 T}\left[\log p + \beta^2(p)\sigma^2 w^2\right]\right).$$

For proof, see Appendix A.4.6

Improvement of $\log p$ factor : The above result can be improved by a $\log p$ factor by considering varying epoch lengths (which depend on the problem parameters). The resulting convergence rate is $\mathcal{O}((s+r)p\log p/T+\alpha^2/p)$. See Theorem 11 in Appendix A.3.

Scaling of $\beta(p)$: We have the following bounds $\Theta(\sqrt{p}) \leq \beta(p)\Theta(p)$. 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 in Section 4.4.2.1.

Comparison with the batch result: Agarwal et al. [2012a] consider the batch version of the same problem (4.2.4), and provide a convergence rate of $\mathcal{O}(s\log p + rp)/T + s\alpha^2/p^2)$. 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 11 are *unimproveable* 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 our work [Sedghi et al., 2014a,b], 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 $s\alpha^2/p^2$ for their batch algorithm. Our online algorithm has an approximation error of $\max\{s + r, p\}\alpha^2/p^2$, which is worse, but is still decaying with p. It is not clear if this bound can be improved by any other online algorithm.

4.4.2.1 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 $\max_{i,j} \mathrm{Var}(N_k(i,j)) = \sigma^2$. 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 w.h.p 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 variable h is hidden, and y is the observed variable. 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. Without loss of generality, we assume that the columns of A are normalized, as 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 $S^* = \sigma_n^2 I$ is a diagonal matrix and $L^* = \sigma_h^2 AA^\top$ has rank at most r.

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 w.h.p.

$$\|n_k n_k^\top - \sigma_n^2 I\|_2 = \mathcal{O}(\sqrt{p}\sigma_n^2), \quad \|h_k h_k^\top - \sigma_h^2 I\|_2 = \mathcal{O}(\sqrt{p}\sigma_h^2). \tag{4.4.6}$$

We thus have with probability $1 - Te^{-cp}$, $\|E_k\|_2 \leq \mathcal{O}\left(\sqrt{p}(\|A\|^2\sigma_h^2 + \sigma_n^2)\right), \forall\, k \leq T$. When $\|A\|_2$ is bounded, we obtain the optimal bound in Theorem 11, 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 $\alpha \sim \text{polylog}(p)$. 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 variable 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 $M^* := \Sigma^{*-1}_{y,y}$, which can be expressed as a combination of sparse and low rank matrices, for the class of latent Gaussian graphical models, described in Section 4.5.3. Note that the result cannot be applied directly in this case as loss function is not locally Lipschitz. Nevertheless, in Section 4.5.3 we show that we can take care of this problem.

4.5 Proof Ideas and Discussion

4.5.1 Proof Ideas for REASON 1

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 Theorem 3.

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 i-th epoch. 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.

Moreover, we show that the parameter ρ_x can be chosen as a constant while the earlier work [Wang and Banerjee, 2013] requires a stronger constraint $\rho_x = \sqrt{T_i}$. For details, see Appendix A.2.1. 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. Note that this is a direct result from local Lipschitz property and it is proved in Lemma 8 in Appendix A.2.1. In fact, in order to prove the guarantees, we need bounded duality which is a weaker assumption than local Lipschitz property. We discuss this in Section 4.5.3.

3. For fixed epoch length, the error shrinkage stops after some epochs but the error does not increase significantly afterwards. Following lines of [Agarwal et al., 2012b], we prove that for this case the convergence rate is worse by a factor of $\log d$.

4.5.2 Proof Ideas for REASON 2

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

   $$\arg\min_{W}\{\lambda_i\phi(W) + \frac{\rho}{2\tau_k}\|W - [W_k + \tau_k A^\top(M_{k+1} - AW_k - \frac{1}{\rho}Z_k)]\|_{\mathbb{F}}^2\}. \quad (4.5.1)$$

   Since the two parts of $W = [S; L]$ are separable in the quadratic part now, Equation (4.5.1) reduces to two decoupled updates on S and L as given by (4.4.2) and (4.4.3).

2. It is convenient to analyze the W update in Equation (4.5.1) to derive convergence rates 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.

It should be noted that we only require LSC condition on $S+L$ and not jointly on S and L. This results in an additional higher order term when analyzing the epoch error and therefore does not play a role in the final convergence bound. The LSC bound provides us with sum of sparse and low rank errors for each epoch. i.e., $\|\hat{S}_i - \bar{S}(T_i) + \hat{L}_i - \bar{L}(T_i)\|_{\mathbb{F}}^2$. Next we need to decouple these errors.

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. These norm constraints allows us to control the error. 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, instead of p.

4. For fixed epoch length, the error shrinkage stops after some epochs but the error does not increase significantly afterwards. Following lines of [Agarwal et al., 2012b], we prove that for this case the convergence rate is worse by a factor of $\log p$.

Thus, our careful analysis leads to tight guarantees for online matrix decomposition. For Proof outline and detailed proof of Theorem 2 see Appendix A.3.1 and A.4 respectively.

4.5.3 Graphical Model Selection

Our framework cannot directly handle the case where loss function is the log likelihood objective. This is because for log likelihood function Lipschitz constant can be large and this leads to loose bounds on error. Yet, as we discuss shortly, our analysis needs conditions weaker than Local Lipschitz property. We consider both settings, i.e., fully observed graphical models and latent Gaussian graphical models. We apply sparse optimization to the former and tackle the latter with sparse + low rank decomposition.

4.5.3.1 Sparse optimization for learning Gaussian graphical models

Consider a p-dimensional Gaussian random vector $[x_1, ..., x_p]^\top$ with a sparse inverse covariance or precision matrix Θ^*. Consider the ℓ_1-regularized maximum likelihood estimator (batch estimate),

$$\widehat{\Theta} := \arg\min_{\Theta \succ 0} \{\mathrm{Tr}(\widehat{\Sigma}\Theta) - \log\det\{\Theta\} + \lambda_n \|\Theta\|_1\}, \tag{4.5.2}$$

where $\widehat{\Sigma}$ 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]. We have that the loss function is strongly convex for all Θ within a ball[1].

However, the above loss function is not (locally) Lipschitz in general, since the gradient[2] $\nabla f(x, \Theta) = xx^\top - \Theta^{-1}$ is not bounded in general. Thus, the bounds derived in Theorem 1 do not directly apply here. However, our conditions for recovery are somewhat weaker than local Lipschitz property, and we provide guarantees for this setting under some additional constraints.

Let $\Gamma^* = \Theta^{*-1} \otimes \Theta^{*-1}$ denote the Hessian of log-determinant barrier at true information matrix. Let $Y_{(j,k)} := X_j X_k - \mathbb{E}[X - jX_k]$ and note that $\Gamma^*_{(j,k),(l,m)} = \mathbb{E}[Y_{(j,k)Y_{(l,m)}}]$ [Ravikumar et al., 2011]. A bound on $|||\Gamma^*|||_\infty$ limits the influence of the

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

[2] 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].

edges on each other, and we need this bound for guaranteed convergence. Yet, this bound contributes to a higher order term and does not show up in the convergence rate.

Corollary 1. *Under Assumptions A1, A2 when the initialization radius R_1 satisfies $R_1 \leq \frac{0.25}{\|\Sigma^*\|_{\mathbb{F}}}$, under the negative log-likelihood loss function, REASON 1 has the following bound (for dual update step size $\tau = \sqrt{T_0}$)*

$$\|\bar{\theta}_T - \theta^*\|_2^2 \leq c_0 \frac{s}{\gamma^2 T} \cdot \frac{\log d}{k_T} \left[\log d + \sigma^2 \left(w^2 + 24 \log(k_T / \log d)\right)\right]$$

The proof does not follow directly from Theorem 1, since it does not utilize Lipschitz property. However, the conditions for Theorem 1 to hold are weaker than (local) Lipschitz property and we utilize it to provide the above result. For proof, see Appendix A.2.7. Note that in case epoch length is not fixed and depends on the problem parameters, the bound can be improved by a $\log d$ factor.

Comparing to Theorem 1, the local Lipschitz constant G^4 is replaced by $\sigma^2 \|\!|\!|\Gamma^*\|\!|\!|^2$. We have $G = \mathcal{O}(d)$, and thus we can obtain better bounds in the above result, when $\|\!|\!|\Gamma^*\|\!|\!|$ is small and the initialization radius R_1 satisfies the above condition. Intuitively, the initialization condition (constraint on R_1) is dependent on the strength of the correlations. For the weak-correlation case, we can initialize with large error compared to the strongly correlated setting.

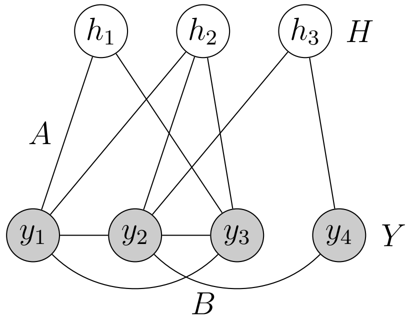


Figure 4.1: Graphical representation of a latent variable model.

4.5.3.2 Sparse + low rank decomposition for learning latent Gaussian graphical models

Consider the Bayesian network on p-dimensional observed variables as

$$y = A\,h + B\,y + n, \quad y, n \in \mathbb{R}^p,\ h \in \mathbb{R}^r, \tag{4.5.3}$$

as in Figure 4.1 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 (4.5.3), the precision matrix M^* with respect to the marginal distribution on the observed vector y is given by

$$M^* := {\Sigma^*}^{-1}_{y,y} = \widetilde{M}^*_{y,y} - \widetilde{M}^*_{y,h}(\widetilde{M}^*_{h,h})^{-1}\widetilde{M}^*_{h,y}, \qquad (4.5.4)$$

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 (4.5.4) has rank at most r. The first term in (4.5.4) 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 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 (4.5.4). 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. Therefore, 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 (4.5.4) 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]

$$\begin{aligned}\{\hat{S}, \hat{L}\} :=& \arg\min\{\mathrm{Tr}(\widehat{\Sigma}_n M) - \log\det M + \lambda_n\|S\|_1 + \mu_n\|L\|_*\}, \\ & s.t. \quad M = S + L.\end{aligned} \tag{4.5.5}$$

This is a special case of (4.2.4) with the loss function $f(M) = \mathrm{Tr}(\widehat{\Sigma}_n\, M) - \log\det M$. 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 (4.4.6).

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 ℓ_1 norm constraint on M as $\|M_k - \tilde{S}_i - \tilde{L}_i\|_1^2 \leq \breve{R}^2$, and this can still be computed efficiently, since it involves projection on to the ℓ_1 norm ball, see Appendix 4.7.1. We assume a good initialization M which satisfies $\|M - M^*\|_1^2 \leq \breve{R}^2$.

This ensures that M_k in subsequent steps is non-singular, and that the gradient of the loss function f in (4.5.5), which involves M_k^{-1}, can be computed. As observed in section 4.5.3.1 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.

There is another reason for using the ℓ_1 bound. Note that the loss function is not generally Lipschitz in this case. However, our conditions for recovery are somewhat weaker than local Lipschitz property, and we provide guarantees for this setting under some additional constraints. Let $\Gamma^* = M^* \otimes M^*$. As explained in Section 4.5.3.1, a bound on $|||\Gamma^*|||_\infty$ limits the influence on the edges on each other, and we need this bound for guaranteed convergence. Yet, this bound contributes to a higher order term and does not show up in the convergence rate.

Corollary 2. *Under Assumptions A1, A2, A4, A5, when the radius $\breve{R}$ satisfies $\breve{R} \leq \frac{0.25}{\|\Sigma^*\|_{\mathbb{F}}}$, under the negative log-likelihood loss function, REASON 2 has the following bound (for dual update step size $\tau = \sqrt{T_0}$)*

$$\|\bar{S}(T) - S^*\|_{\mathbb{F}}^2 + \|\bar{L}(T) - L^*\|_{\mathbb{F}}^2 \leq$$
$$\frac{c_0(s+r)}{T} \cdot \frac{\log p}{k_T} \left[\log p + \beta^2(p)\sigma^2 \left(w^2 + \log(k_T/\log d)\right)\right] + \max\{s+r,p\}\frac{\alpha^2}{p}.$$

The proof does not follow directly from Theorem 2, since it does not utilize Lipschitz property. However, the conditions for Theorem 2 to hold are weaker than (local) Lipschitz property and we utilize it to provide the above result. For proof, see Appendix A.4.7. Note that in case epoch length is not fixed and depends on the problem parameters, the bound can be improved by a $\log p$ factor.

4.6 Experiments

4.6.1 REASON 1

For sparse optimization problem we compare REASON 1 with RADAR and ST-ADMM under the least-squares regression setting. Samples (x_t, y_t) are generated such that $x_t \in \text{Unif}[-B, B]$ and $y_t = \langle \theta^*, x \rangle + n_t$. θ^* is s-sparse with $s = \lceil \log d \rceil$. $n_t \sim \mathcal{N}(0, \eta^2)$. With $\eta^2 = 0.5$ in all cases. We consider $d = 20, 2000, 20000$ and $s = 1, 3, 5$ respectively. The experiments are performed on a 2.5 GHz Intel Core i5 laptop with 8 GB RAM. See Table 4.3 for experiment results. It should be noted that

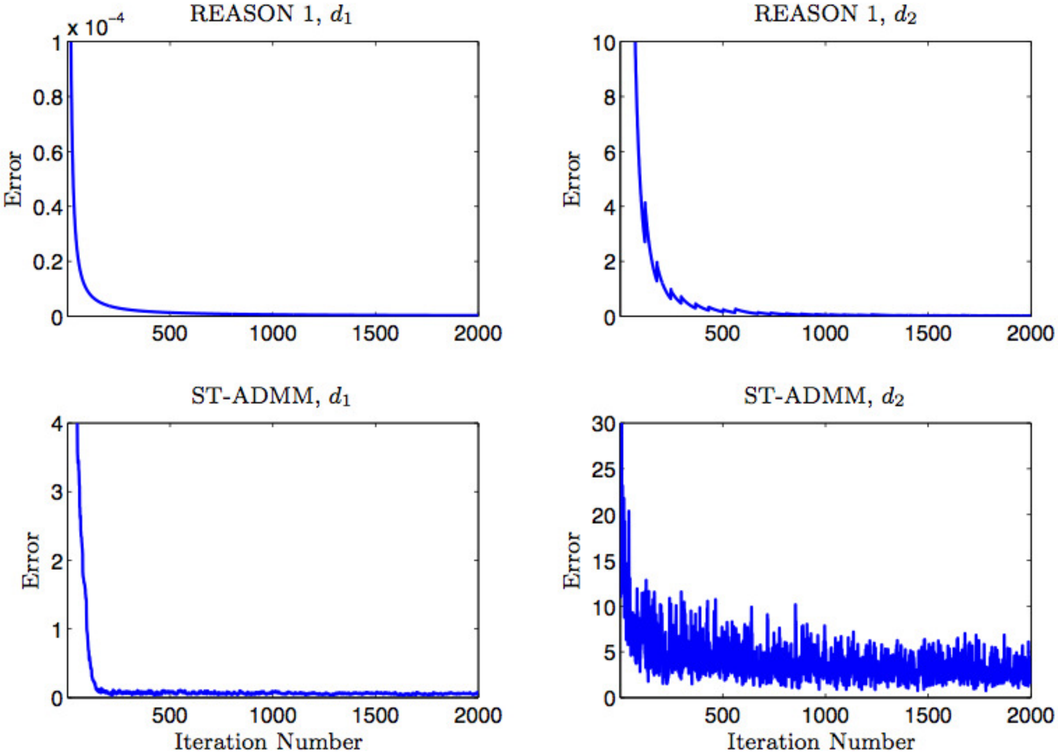


Figure 4.2: Least square regression, Error= $\frac{\|\theta-\theta^*\|_2}{\|\theta^*\|_2}$ vs. iteration number, $d_1 = 20$ and $d_2 = 20000$.

RADAR is provided with information of θ^* for epoch design and recentering. In addition, both RADAR and REASON 1 have the same initial radius. Nevertheless, REASON 1 reaches better accuracy within the same run time even for small time frames. In addition, we compare relative error $\|\theta - \theta^*\|_2/\|\theta^*\|_2$ in REASON 1 and ST-ADMM in the first epoch. We observe that in higher dimension error fluctuations for ADMM increases noticeably (see Figure 4.2). Therefore, projections of REASON 1 play an important role in denoising and obtaining good accuracy.

Epoch Size For fixed- epoch size, if epoch size is designed such that the relative error defined above has shrunk to a stable value, then we move to the next epoch

Dimension	Run Time (s)	Method	error at 0.02T	error at 0.2T	error at T
d=20000	T=50	ST-ADMM	1.022	1.002	0.996
		RADAR	0.116	2.10e-03	6.26e-05
		REASON 1	1.5e-03	2.20e-04	1.07e-08
d=2000	T=5	ST-ADMM	0.794	0.380	0.348
		RADAR	0.103	4.80e-03	1.53e-04
		REASON 1	0.001	2.26e-04	1.58e-08
d=20	T=0.2	ST-ADMM	0.212	0.092	0.033
		RADAR	0.531	4.70e-03	4.91e-04
		REASON 1	0.100	2.02e-04	1.09e-08

Table 4.3: *Least square regression problem, epoch size* $T_i = 2000$*, Error*$= \frac{\|\theta-\theta^*\|_2}{\|\theta^*\|_2}$.

Run Time	$T = 50$ sec			$T = 150$ sec		
Error	$\frac{\|M^*-S-L\|_{\mathbb{F}}}{\|M^*\|_{\mathbb{F}}}$	$\frac{\|S-S^*\|_{\mathbb{F}}}{\|S^*\|_{\mathbb{F}}}$	$\frac{\|L^*-L\|_{\mathbb{F}}}{\|L^*\|_{\mathbb{F}}}$	$\frac{\|M^*-S-L\|_{\mathbb{F}}}{\|M^*\|_{\mathbb{F}}}$	$\frac{\|S-S^*\|_{\mathbb{F}}}{\|S^*\|_{\mathbb{F}}}$	$\frac{\|L^*-L\|_{\mathbb{F}}}{\|L^*\|_{\mathbb{F}}}$
REASON 2	2.20e-03	0.004	0.01	5.55e-05	1.50e-04	3.25e-04
inexact ALM	5.11e-05	0.12	0.27	8.76e-09	0.12	0.27

Table 4.4: *REASON 2 and inexact ALM, matrix decomposition problem.* $p = 2000$, $\eta^2 = 0.01$

and the algorithm works as expected. If we choose a larger epoch than this value we do not gain much in terms of accuracy at a specific iteration. On the other hand if we use a small epoch size such that the relative error is still noticeable, this delays the error reduction and causes some local irregularities.

4.6.2 REASON 2

We compare REASON 2 with state-of-the-art inexact ALM method for matrix decomposition problem[3] In this problem M is the noisy sample the algorithm receives. Since we have direct access to M, the M-update is eliminated.

Table 4.4 shows that with equal time, inexact ALM reaches smaller $\frac{\|M^*-S-L\|_{\mathbb{F}}}{\|M^*\|_{\mathbb{F}}}$ error while in fact this does not provide a good decomposition. On the other hand, REASON 2 reaches useful individual errors in the same time frame. Experiments with $\eta^2 \in [0.01, 1]$ reveal similar results. This emphasizes the importance of projections in REASON 2. Further investigation on REASON 2 shows that performing one of the projections (either ℓ_1 or nuclear norm) suffices to reach this performance. The same precision can be reached using only one of the projections. Addition of the second projection improves the performance marginally. Performing nuclear norm projections are much more expensive since they require SVD. Therefore, it is more efficient to perform the ℓ_1 projection. Similar experiments on exact ALM shows worse performance than inexact ALM and are thus omitted.

4.7 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.

[3] ALM codes are downloaded from http://perception.csl.illinois.edu/matrix-rank/home.html and REASON 2 code is available at https://github.com/haniesedghi/REASON2.

4.7.1 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

$$\begin{aligned}
&\min_{w} \ \|w - v\|_2^2 \ \ s.t. \ \ \|w\|_1 \leq R, \qquad (4.7.1)\\
&w = \theta - \tilde{\theta}_i,\\
&R = R_i,\\
&v = \frac{1}{\rho + \rho_x}[y_k - \tilde{\theta}_i - \frac{f(\theta_k)}{\rho} + \frac{z_k}{\rho} + \frac{\rho_x}{\rho}(\theta_k - \tilde{\theta}_i)].
\end{aligned}$$

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 4 for solving (4.7.1). The complexity of this Algorithm is $\mathcal{O}(d \log d)$, $d = p^2$.

4.7.2 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 (4.7.1) with

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

$$W = S - \tilde{S}_i, \quad R = R_i.$$

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 4. In other words, at each iteration,

$$\text{vector}\left(W^{(t)} - \eta_t\left[\lambda_i \nabla^{(t)}\|W^{(t)} + \tilde{S}_i\|_1 + \frac{\rho}{\tau_k}(W^{(t)} - (S_k + \tau_k G_{M_k} - \tilde{S}_i))\right]\right)$$

is the input to Algorithm 4 (instead of vector v) and the output is 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 is Algorithm 5. 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.

The code for REASON 1 follows directly from the discussion in Section 4.7.1. For REASON 2 on the other hand, we have added additional heuristic modifications to improve the performance. REASON 2 code is available at https://github.com/haniesedghi/REASON2. The first modification is that we do not update the dual variable Z per every iteration on S and L. Instead, we update the dual variable once S and L seem to have converged to some value or after every m iterations on S and L. The reason is that once we start the iteration, S and L can be far from each other which results in a big dual variable and hence, a slower convergence. The value of m can be set based on the problem. For the experiments discussed here we have used $m = 4$.

Further investigation on REASON 2 shows that performing one of the projections (either ℓ_1 or nuclear norm) suffices to reach this performance. The same precision can be reached using only one of the projections. Addition of the second projection improves the performance only marginally. Performing nuclear norm projections are much more expensive since they require SVD. Therefore, it is more efficient to perform the ℓ_1 projection. In the code, we leave it as an option to run both projections.

4.8 Conclusion

In our work [Sedghi et al., 2014a,b], 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 the sparse optimization problem, we showed that we reach the minimax-optimal rate in this case, which implies that our guarantee is unimproveable by any (batch or online) algorithm (up to constant factors). We then consider the matrix decomposition problem into sparse and low rank components, and propose a modified version of the multi-block ADMM algorithm. Experiments show that for both sparse optimization and matrix decomposition problems, our algorithm outperforms the state-of-the-art methods. In particular, we reach higher accuracy with same time complexity. There are various future problems to consider. One is to provide lower bounds on error for matrix decomposition problem in case of strongly convex loss if possible. Agarwal et al. [2012a] do not provide bounds for strongly convex functions. Another approach can be to extend our method to address nonconvex programs. Loh and Wainwright [2013] and Wang et al. [2013c] show that if the problem is nonconvex but has additional properties, it can be solved by methods similar to convex loss programs. In addition, we can extend our method to coordinate descent methods such as [Roux et al., 2012].

Algorithm 3 Regularized Epoch-based Admm for Stochastic Optimization in high-dimensioN 2 (REASON 2)

```
Input $\rho, \rho_x > 0$, epoch length $T_0$ , 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 (4.3.3), $G_{M_k} = M_{k+1} - S_k - L_k - \frac{1}{\rho} Z_k$.

for Each epoch $i = 1, 2, ..., k_T$ do
    Initialize $S_0 = \tilde{S}_i, L_0 = \tilde{L}_i, M_0 = S_0 + L_0$
    for Each iteration $k = 0, 1, ..., T_0 - 1$ do
        $M_{k+1} = \frac{-\nabla f(M_k) + Z_k + \rho(S_k + L_k) + \rho_x M_k}{\rho + \rho_x}$
        $S_{k+1} = \min_{\|S - \tilde{S}_i\|_1 \leq R_i} \lambda_i \|S\|_1 + \frac{\rho}{2\tau_k} \|S - (S_k + \tau_k G_{M_k})\|_{\mathbb{F}}^2$
        $L_{k+1} = \min_{\|L - \tilde{L}_i\|_* \leq \tilde{R}_i} \mu_i \|L\|_* + \frac{\rho}{2} \|L - Y_k - U_k/\rho\|_{\mathbb{F}}^2$
        $Y_{k+1} = \min_{\|Y\|_\infty \leq \alpha/p} \frac{\rho}{2\tau_k} \|Y - (L_k + \tau_k G_{M_k})\|_{\mathbb{F}}^2 + \frac{\rho}{2} \|L_{k+1} - Y - U_k/\rho\|_{\mathbb{F}}^2$
        $Z_{k+1} = Z_k - \tau(M_{k+1} - (S_{k+1} + L_{k+1}))$
        $U_{k+1} = U_k - \tau(L_{k+1} - Y_{k+1}).$
    end for
    Set: $\tilde{S}_{i+1} = \frac{1}{T_0} \sum_{k=0}^{T_0 - 1} S_k$ and $\tilde{L}_{i+1} := \frac{1}{T_0} \sum_{k=0}^{T_0 - 1} L_k$
    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
```

Algorithm 4 Implementation of θ-update

```
Input: A vector $v = \frac{1}{\rho+\rho_x}[y_k - \tilde{\theta}_i - \frac{\nabla f(\theta_k)}{\rho} + \frac{z_k}{\rho} + \frac{\rho_x}{\rho}(\theta_k - \tilde{\theta}_i)]$ and a scalar $R = R_i > 0$

if $\|v\|_1 \leq R$, then
    Output: $\theta = v + \tilde{\theta}_i$
else
    Sort $v$ into $\mu$: $\mu_1 \geq \mu_2 \geq \cdots \geq \mu_d$.
    Find $\kappa = \max\{j \in [d] : \mu_j - \frac{1}{j}\big(\sum_{i=1}^{j} \mu_i - R\big) > 0\}$.
    Define $\zeta = \frac{1}{\kappa}\big(\sum_{i=1}^{\kappa} \mu_i - R\big)$
    Output: $\theta$, where $\theta_{(j)} = \text{sign}(v_{(j)}) \max\{v_{(j)} - \zeta, 0\} + (\tilde{\theta}_i)_{(j)}$
end if
```

Algorithm 5 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
    $v = W^{(t)} - \eta_t \Big[\lambda_i \nabla^{(t)} \|W^{(t)} + \text{vector}(\tilde{S}_i)\|_1 + \frac{\rho}{\tau_k}\Big(W^{(t)} - \text{vector}(S_k + \tau_k G_{M_k} - \tilde{S}_i)\Big)\Big]$

    if $\|v\|_1 \leq R$, then
        $W^{(t+1)} = v$
    else
        Sort $v$ into $\mu$: $\mu_1 \geq \mu_2 \geq \cdots \geq \mu_d$.
        Find $\kappa = \max\{j \in [d] : \mu_j - \frac{1}{j}\big(\sum_{i=1}^{j} \mu_i - R\big) > 0\}$.
        Define $\zeta = \frac{1}{\kappa}\big(\sum_{i=1}^{\kappa} \mu_i - R\big)$
        For $1 \leq j \leq d$, $W^{(t+1)}_{(j)} = \text{sign}(v_{(j)}) \max\{v_{(j)} - \zeta, 0\}$
    end if
end for
Output:matrix($W^{(t_s)}$) $+ \tilde{S}_i$
```

Appendix A

Appendix: Proofs

A.1 Guarantees for REASON 1

First, we provide guarantees for the theoretical case such that epoch length depends on epoch radius. This provides intuition on how the algorithm is designed. The fixed-epoch algorithm is a special case of this general framework. We first state and prove guarantees for general framework. Next, we leverage these results to prove Theorem 1.

Let the design parameters be set as

$$\begin{aligned}
T_i &= C\frac{s^2}{\gamma^2}\left[\frac{\log d + 12\sigma_i^2\log(3/\delta_i)}{R_i^2}\right], \qquad \text{(A.1.1)}\\
\lambda_i^2 &= \frac{\gamma}{s\sqrt{T_i}}\sqrt{R_i^2\log d + \frac{G^2R_i^2 + \rho_x^2R_i^4}{T_i} + \sigma_i^2R_i^2\log(3/\delta_i)},\\
\rho &\propto \frac{\sqrt{\log d}}{R_i\sqrt{T_i}}, \quad \rho_x > 0, \quad \tau = \rho.
\end{aligned}$$

Theorem 3. *Under assumptions $A1 - A3$ and parameter settings* (A.1.1)*, there exists a constant $c_0 > 0$ such that REASON 1 satisfies for all $T > k_T$,*

$$\|\bar{\theta}_T - \theta^*\|_2^2 \le c_0 \frac{s}{\gamma^2 T}\left[e \log d + \sigma^2 w^2 + \log k_T)\right], \tag{A.1.2}$$

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

For Proof outline and detailed proof of Theorem 3 see Appendix A.1.1 and A.2 respectively.

A.1.1 Proof outline for Theorem 3

The foundation block for this proof is Proposition 2.

Proposition 2. *Suppose f satisfies Assumptions $A1, A2$ with parameters γ and σ_i respectively and assume that $\|\theta^* - \tilde{\theta}_i\|_1^2 \le R_i^2$. We apply the updates in REASON 1 with parameters as in* (A.1.1)*. Then, there exists a universal constant c such that for any radius R_i*

$$f(\bar{\theta}(T_i)) - f(\hat{\theta}_i) + \lambda_i \|\bar{y}(T_i)\|_1 - \lambda_i \|\hat{\theta}_i\|_1 \le \frac{R_i\sqrt{\log d}}{\sqrt{T_i}} + \frac{G R_i}{T_i} + \frac{\rho_x R_i^2}{T_i} \tag{A.1.3a}$$

$$+ \frac{R_i \sigma_i}{\sqrt{T_i}}\sqrt{12 \log(3/\delta_i)},$$

$$\|\bar{\theta}(T_i) - \theta^*\|_1^2 \le \frac{c'}{\sqrt{C}} R_i^2. \tag{A.1.3b}$$

where $\rho_0 = \rho_x + \rho$ and both bounds are valid with probability at least $1 - \delta_i$.

Note that our proof for epoch optimum improves proof of [Wang and Banerjee, 2013] with respect to ρ_x. For details, see Section A.2.1.

In order to prove Proposition 2, 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 4. *At epoch i assume that $\|\theta^* - \tilde{\theta}_i\|_1 \leq R_i$. Then the error Δ_i satisfies the bounds*

$$\|\hat{\theta}_i - \theta^*\|_2 \leq \frac{4}{\gamma}\sqrt{s}\lambda_i, \tag{A.1.4a}$$

$$\|\hat{\theta}_i - \theta^*\|_1 \leq \frac{8}{\gamma}s\lambda_i. \tag{A.1.4b}$$

Lemma 5. *Under the conditions of Proposition 2 and with parameter settings* (A.1.1) *, we have*

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

with probability at least $1 - \delta_i$.

A.2 Proof of Theorem 3

The first step is to ensure that $\|\theta^* - \tilde{\theta}_i\| \leq R_i$ holds at each epoch so that Proposition 2 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 (A.1.1) where $C \geq 1$ is a constant we can choose. By substituting this T_i in inequality (A.1.3b), the simplified bound (A.1.3b) 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 5 applies and we find that

$$\|\hat{\Delta}(T_i)\|_2^2 \leq \frac{c}{s} R_i^2. \tag{A.2.1}$$

We have now bounded $\hat{\Delta}(T_i) = \bar{\theta}(T_i) - \hat{\theta}_i$ and Lemma 4 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 (A.1.4a) with (A.2.1), we get

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

i.e.

$$\|\Delta^*(T_i)\|_2^2 \leq c\frac{R_1^2 2^{-(i-1)}}{s}. \tag{A.2.2}$$

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 $w_i^2 = w^2 + 24\log i$, we can apply union bound to simplify the error probability as $1 - 6\exp(-w^2/12)$. Throughout this report 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 (A.2.2) from its dependence on the number of epochs k_T to the number of iterations needed to complete k_T epochs, i.e. $T(K) = \sum_{i=1}^{k} T_i$. Note that here we use T_i from (A.2.8), to show that when considering the dominant terms, the definition in (A.1.1) suffices. Here you can see how negligible terms are ignored.

$$\begin{aligned} T(k) &= \sum_{i=1}^{k} C\left[\frac{s^2}{\gamma^2}\left[\frac{\log d + 12\sigma_i^2\log(3/\delta_i)}{R_i^2}\right] + \frac{s}{\gamma}\frac{G}{R_i} + \frac{s}{\gamma}\rho_x\right] \\ &= C\sum_{i=1}^{k}\left[\frac{s^2\{\log d + \gamma/sG + \sigma^2(w^2 + 24\log k)\}2^{i-1}}{\gamma^2 R_1^2} + \frac{sG}{\gamma R_1}\sqrt{2}^{i-1} + \frac{s}{\gamma}\rho_x\right]. \end{aligned}$$

Hence,

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

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

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

Ignoring the dominated terms and using a first order approximation for $\log(a+b)$,

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

Therefore,

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

Putting this back into (A.2.2), we get that

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

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 (A.1.2).

A.2.1 Proofs for Convergence within a Single Epoch for Algorithm 2

Lemma 6. *For $\bar{\theta}(T_i)$ defined in Algorithm 2 and $\hat{\theta}_i$ the optimal value for epoch i, let $\rho = c_1\sqrt{T_i}$, ρ_x some positive constant, $\rho_0 = \rho + \rho_x$ and $\tau = \rho$ where $c_1 = \frac{\sqrt{\log d}}{R_i}$. We have that*

$$\begin{aligned}&f(\bar{\theta}(T_i)) - f(\hat{\theta}_i) + \lambda_i\|\bar{y}(T_i)\|_1 - \lambda_i\|\hat{\theta}_i\|_1 \leq \\ &\frac{R_i\sqrt{\log d}}{\sqrt{T_i}} + \frac{GR_i}{T_i} + \frac{\rho_x R_i^2}{T_i} + \frac{\sum_{k=1}^{T_i}\langle e_k, \hat{\theta}_i - \theta_k\rangle}{T_i}.\end{aligned} \tag{A.2.3}$$

Remark : Please note that as opposed to [Wang and Banerjee, 2013] we do not require $\rho_x \propto \sqrt{T_i}$. We show that our parameter setting also works.

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,

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

As a result, we can write down the update rule of θ in REASON 1 as

$$\begin{aligned} \theta_{k+1} = \underset{\|\theta - \tilde{\theta}_i\|_1^2 \leq R_i^2}{\arg\min} \{ & f(\theta) - B_f(\theta, \theta_k) + z_k^T(\theta - y_k) + \rho B_\phi(\theta, y_k) \\ & + \rho_x B_{\phi'_x}(\theta, \theta_k)\}. \end{aligned}$$

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

$$\theta_{k+1} = \underset{\|\theta - \tilde{\theta}_i\|_1^2 \leq R_i^2}{\arg\min} \{f(\theta) + \langle z_k, \theta - y_k \rangle + \rho B_\phi(\theta, y_k) + \rho_x B_{\phi_x}(\theta, \theta_k)\}. \tag{A.2.4}$$

We notice that equation (A.2.4) is equivalent to Equation (7) [Wang and Banerjee, 2013]. Note that as opposed to [Wang and Banerjee, 2013], in our setting ρ_x can be set as a constant. Therefore, for completeness we provide proof of convergence and the convergence rate for our setting.

Lemma 7. *Convergence of REASON 1: The optimization problem defined in REASON 1 converges.*

Proof. On lines of [Wang and Banerjee, 2013], let $\mathbf{R}(k+1)$ stand for residuals of optimality condition. For convergence we need to show that $\lim_{k\to\infty} \mathbf{R}(k+1) = 0$. Let $w_k = (\theta_k, y_k, z_k)$. Define

$$D(w^*, w_k) = \frac{1}{\tau\rho}\|z^* - z_k\|_2^2 + B_\phi(y^*, y_k) + \frac{\rho_x}{\rho} B_\phi(\theta^*, \theta_k).$$

By Lemma 2 Wang and Banerjee [2013]

$$\mathbf{R}(t+1) \leq D(w^*, w_k) - D(w^*, w_{k+1}).$$

Therefore,

$$\begin{aligned}\sum_{k=1}^{\infty} \mathbf{R}(t+1) &\leq D(w^*, w_0) \\ &= \frac{1}{\tau\rho}\|z^*\|_2^2 + B_\phi(y^*, y_0) + \frac{\rho_x}{\rho} B_\phi(\theta^*, \theta_0) \\ &\leq \lim_{T\to\infty} \frac{R_i^2}{\log d\ T}\|\nabla f(\theta^*)\|_2^2 + 2R_i^2 + \frac{\rho_x}{\sqrt{T\log d}} R_i^3.\end{aligned}$$

Therefore, $\lim_{k\to\infty} \mathbf{R}(k+1) = 0$ and the algorithm converges. ■

If in addition we incorporate sampling error, then Lemma 1 [Wang and Banerjee, 2013] changes to

$$
\begin{aligned}
& f(\theta_{k+1}) - f(\hat{\theta}_i) + \lambda_i \|y_{k+1}\|_1 - \lambda_i \|\hat{\theta}_i\|_1 \leq \\
& - \langle z_k, \theta_{k+1} - y_{k+1} \rangle - \frac{\rho}{2}\{\|\theta_{k+1} - y_k\|_2^2 + \|\theta_{k+1} - y_{k+1}\|_2^2\} + \langle e_k, \hat{\theta}_i - \theta_k \rangle \\
& + \frac{\rho}{2}\{\|\hat{\theta}_i - y_k\|_2^2 - \|\hat{\theta}_i - y_{k+1}\|_2^2\} + \rho_x\{B_{\phi_x}(\hat{\theta}_i, \theta_k) - B_{\phi_x}(\hat{\theta}_i, \theta_{k+1}) \\
& - B_{\phi_x}(\theta_{k+1}, \theta_k)\}.
\end{aligned}
$$

The above result follows from convexity of f, the update rule for θ (Equation (A.2.4)) and the three point property of Bregman divergence.

Next, we show the bound on the dual variable.

Lemma 8. *The dual variable in REASON 1 is bounded. i.e.,*

$$\|z_k\|_1 \leq G + 2\rho_0 R_i, \quad \textit{where} \quad \rho_0 := \rho_x + \rho.$$

Proof. Considering the update rule for θ, we have the Lagrangian

$$\mathcal{L} = f(\theta) + \langle z_k, \theta - y_k \rangle + \rho B_\phi(\theta, y_k) + \rho_x B_{\phi_x}(\theta, \theta_k) + \zeta \left(\|\theta_{k+1} - \tilde{\theta}_i\|_1 - R_i \right),$$

where ζ is the Lagrange multiplier corresponding to the ℓ_1 bound. We hereby emphasize that ζ does not play a role in size of the dual variable. i.e., considering the ℓ_1 constraint, three cases are possible:

1. $\|\theta_{k+1} - \tilde{\theta}_i\|_1 > R_i$. By complementary slackness, $\zeta = 0$.

2. $\|\theta_{k+1} - \tilde{\theta}_i\|_1 < R_i$. By complementary slackness, $\zeta = 0$.

3. $\|\theta_{k+1} - \tilde{\theta}_i\|_1 = R_i$. This case is equivalent to the non-constrained update and no projection will take place. Therefore, z will be the same as in the non-constrained update.

Having above analysis in mind, the upper bound on the dual variable can be found as follows By optimality condition on θ_{k+1}, we have

$$-z_k = \nabla f(\theta_{k+1}) + \rho_x(\theta_{k+1} - \theta_k) + \rho(\theta_{k+1} - y_k). \tag{A.2.5}$$

By definition of the dual variable and the fact that $\tau = \rho$, we have that

$$z_k = z_{k-1} - \rho(\theta_k - y_k)$$

Hence, we have that $-z_{k-1} = \nabla f(\theta_{k+1}) + (\rho_x + \rho)(\theta_{k+1} - \theta_k)$. Therefore,

$$\|z_{k-1}\|_1 \leq G + 2\rho_0 R_i, \quad \text{where} \quad \rho_0 := \rho_x + \rho.$$

It is easy to see that this is true for all z_k at each epoch. ■

Consequently,

$$\begin{aligned}\frac{-1}{\tau}\langle z_k, z_k - z_{k+1}\rangle &= \frac{1}{\tau}\langle 0 - z_k, z_k - z_{k+1}\rangle \\ &= \frac{1}{2\tau}\left(\|z_{k+1}\|^2 - \|z_k\|^2 - \|z_{k+1} - z_k\|^2\right).\end{aligned}$$

Ignoring the negative term in the upper bound and noting $z_0 = 0$, we get

$$\begin{aligned}\frac{1}{T_i}\sum_{k=1}^{T_i} -\langle z_k, \theta_{k+1} - y_{k+1}\rangle \leq \frac{1}{2\tau T_i}\|z_{T_i}\|^2 &\leq \frac{1}{2\tau T_i}(G + 2\rho_0 R_i)^2 \\ &\simeq \frac{R_i\sqrt{\log d}}{\sqrt{T_i}} + \frac{GR_i}{T_i}.\end{aligned}$$

Note that since we consider the dominating terms in the final bound, terms with higher powers of T_i can be ignored throughout the proof. Next, following the same approach as in Theorem 4 [Wang and Banerjee, 2013] and considering the sampling error, we get,

$$\begin{aligned}&f(\bar{\theta}(T_i)) - f(\hat{\theta}_i) + \lambda_i\|\bar{y}(T_i)\|_1 - \lambda_i\|\hat{\theta}_i\|_1 \\ &\leq \frac{R_i\sqrt{\log d}}{\sqrt{T_i}} + \frac{GR_i}{T_i} + \frac{c_1}{\sqrt{T_i}}\|\hat{\theta}_i - y_0\|_2^2 + \frac{\rho_x}{T_i}B_{\phi_x}(\hat{\theta}_i, \theta_0) + \frac{1}{T_i}\sum_{k=1}^{T_i}\langle e_k, \hat{\theta}_i - \theta_k\rangle.\end{aligned}$$

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,

$$
\begin{aligned}
& f(\bar{\theta}(T_i)) - f(\hat{\theta}_i) + \lambda_i \|\bar{y}(T_i)\|_1 - \lambda_i \|\hat{\theta}_i\|_1 \\
& \leq \frac{R_i \sqrt{\log d}}{\sqrt{T_i}} + \frac{G R_i}{T_i} + \frac{c_1}{\sqrt{T_i}} \|\hat{\theta}_i - \tilde{\theta}_i\|_2^2 + \frac{\rho_x}{T_i} \{B_{\phi'_x}(\hat{\theta}_i, \tilde{\theta}_i) - B_f(\hat{\theta}_i, \tilde{\theta}_i)\} + \sum_{k=1}^{T_i} \langle e_k, \hat{\theta}_i - \theta_k \rangle \\
& \leq \frac{R_i \sqrt{\log d}}{\sqrt{T_i}} + \frac{G R_i}{T_i} + \frac{\sqrt{\log d}}{R_i \sqrt{T_i}} \|\hat{\theta}_i - \tilde{\theta}_i\|_2^2 + \frac{\rho_x}{T_i} B_{\phi'_x}(\hat{\theta}_i, \tilde{\theta}_i) + \sum_{k=1}^{T_i} \langle e_k, \hat{\theta}_i - \theta_k \rangle.
\end{aligned}
$$

We note that $\rho_x B_{\phi'_x}(\hat{\theta}_i, \tilde{\theta}_i) = \frac{\rho_x}{2} \|\hat{\theta}_i - \tilde{\theta}_i\|_2^2$.

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

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

■

A.2.2 Proof of Proposition 2: Inequality (A.1.3a)

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 6, we have that

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

Using Lemma 7 from [Agarwal et al., 2012b], we have that

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

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

A.2.3 Proof of Lemma 4

Proof follows the same approach as Lemma 1 [Agarwal et al., 2012b]. Note 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.

A.2.4 Proof of Lemma 5

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

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

with probability at least $1 - \delta_i$.

A.2.5 Proof of Proposition 2: Inequality (A.1.3b)

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 (A.1.3a) from function values into ℓ_1 and ℓ_2-norm bounds by exploiting the sparsity of θ^*. Since the error bound in (A.1.3a) holds for the minimizer $\hat{\theta}_i$, it also holds for any other feasible vector. In particluar, applying it to θ^* leads to,

$$\begin{aligned}&f(\bar{\theta}(T_i)) - f(\theta^*) + \lambda_i\|\bar{y}(T_i)\|_1 - \lambda_i\|\theta^*\|_1 \\ &\leq \frac{R_i\sqrt{\log d}}{\sqrt{T_i}} + \frac{GR_i}{T_i} + \frac{\rho_x R_i^2}{T_i} + \frac{R_i\sigma_i}{\sqrt{T_i}}\sqrt{12\log\frac{3}{\delta_i}},\end{aligned}$$

with probability at least $1 - \delta_i$.

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

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

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

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

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

$$\begin{aligned}
\|\bar{\theta}(T_i) - \bar{y}(T_i)\|_1 &= \|\frac{1}{T_i}\sum_{k=0}^{T_i-1}(\theta_k - y_k)\|_1 \\
&= \|\frac{1}{\tau T_i}\sum_{k=0}^{T_i-1}(z_{k+1} - z_k)\|_1 \\
&= \frac{1}{\tau T_i}\|z_{T_i}\|_1 \\
&\leq \frac{G + 2\rho_0 R_i}{T_i\tau} = \frac{GR_i}{T_i\sqrt{T_i}\sqrt{\log d}} + \frac{R_i}{T_i}.
\end{aligned}$$

By triangle inequality

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

Hence, after ignoring the dominated terms,

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

By Lemma 6 in [Agarwal et al., 2012b],

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

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

We have $\Delta^*(T_i) - \hat{\Delta}(T_i) = \hat{\theta}_i - \theta^*$. 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 (A.1.4b), we get

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

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

$$\|\hat{\Delta}(T_i)\|_1^2 \leq 4s\|\hat{\Delta}(T_i)\|_2^2 + c\left[\frac{s^2\lambda_i^2}{\gamma^2} + \frac{R_i^2 \log d}{\lambda_i^2 T_i} + \frac{G^2 R_i^2}{\lambda_i^2 T_i^2} + \frac{\rho_x^2 R_i^4}{\lambda_i^2 T_i^2} + \frac{12 R_i^2 \sigma_i^2 \log(\frac{3}{\delta_i})}{T_i \lambda_i^2} + \frac{R_i^2}{T_i^2}\right], \tag{A.2.6}$$

with probability at least $1 - \delta_i$.

Optimizing the above bound with choice of λ_i gives us (A.1.1). From here on all equations hold with probability at least $1 - \delta_i$, we have

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

Thus, for some other c, we have that

$$\|\hat{\Delta}(T_i)\|_1^2 \leq c\frac{s}{\gamma}\left[\frac{R_i\sqrt{\log d}}{\sqrt{T_i}} + \frac{GR_i}{T_i} + \frac{\rho_x R_i^2}{T_i} + \frac{R_i\sigma_i}{\sqrt{T_i}}\sqrt{12\log(\frac{3}{\delta_i})}\right] + \frac{R_i^2}{T_i^2}. \tag{A.2.7}$$

Combining the above inequality with error bound (A.1.4b) for $\hat{\theta}_i$ and using triangle inequality leads to

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

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 (A.2.7). 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

$$T_i = C \left[\frac{s^2}{\gamma^2} \left[\frac{\log d + 12\sigma_i^2 \log(3/\delta_i)}{R_i^2} \right] + \frac{sG}{\gamma R_i} + \frac{s}{\gamma} \rho_x \right], \tag{A.2.8}$$

for a suitably large universal constant C. Note that since we consider the dominating terms in the final bound, the last two terms can be ignored. By design of T_i, we have that

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

which completes this proof.

A.2.6 Proof of Guarantees with Fixed Epoch Length, Sparse Case

This is a special case of Theorem 3 (Appendix). The key difference between this case and optimal epoch length setting of Theorem 3 is that in the latter we guaranteed error halving by the end of each epoch whereas with fixed epoch length that statement may not be possible after the number of epochs becomes large enough. Therefore, we need to show that in such case the error does not increase much to invalidate our analysis. Let k^* be the epoch number such that error halving holds true until then. Next we demonstrate that error does not increase much for $k > k^*$.

Given a fixed epoch length $T_0 = \mathcal{O}(\log d)$, we define

$$k^* := \sup \left\{ i : 2^{j/2+1} \leq \frac{cR_1\gamma}{s} \sqrt{\frac{T_0}{\log d + \sigma_i^2 w^2}} \text{ for all epochs } \; j \leq i \right\}, \tag{A.2.9}$$

where $w = \log(6/\delta)$.

First we show that if we run REASON 1 with fixed epoch length T_0 it has error halving behavior for the first k^* epochs.

Lemma 9. *For $T_0 = \mathcal{O}(\log d)$ and k^* as in (A.2.9), we have*

$$\|\tilde{\theta}_k - \theta^*\|_1 \leq R_k \quad \textit{and} \quad \|\tilde{\theta}_k - \bar{\theta}_k\|_1 \leq R_k \quad \textit{for all} \quad 1 \leq k \leq k^* + 1.$$

with probability at least $1 - 3k\exp(-w^2/12)$. *Under the same conditions, there exists a universal constant* c *such that*

$$\|\tilde{\theta}_k - \theta^*\|_2 \leq c\frac{R_k}{\sqrt{s}} \quad \textit{and} \quad \|\tilde{\theta}_k - \bar{\theta}_k\|_2 \leq c\frac{R_k}{\sqrt{s}} \quad \textit{for all} \;\; 2 \leq k \leq k^* + 1.$$

Next, we analyze the behavior of REASON 1 after the first k^* epochs. Since we cannot guarantee error halving, we can also not guarantee that θ^* remains feasible at later epochs. We use Lemma 10 to control the error after the first k^* epochs.

Lemma 10. *Suppose that Assumptions* $A1 - A3$ *in the main text are satisfied at epochs* $i = 1, 2, \ldots$. *Assume that at some epoch* k, *the epoch center* $\tilde{\theta}_k$ *satisfies the bound* $\|\tilde{\theta}_k - \theta^*\|_2 \leq c_1 R_k/\sqrt{s}$ *and that for all epochs* $j \geq k$, *the epoch lengths satisfy the bounds*

$$\frac{s}{\gamma}\sqrt{\frac{\log d + \sigma_i^2 w_i^2}{T_j}} \leq \frac{R_k}{2} \quad \textit{and} \quad \frac{\log d}{T_i} \leq c_2.$$

Then for all epochs $j \geq k$, *we have the error bound* $\|q_j - \theta^*\|_2^2 \leq c_2 \frac{R_k^2}{s}$ *with probability at least* $1 - 3\sum_{i=k+1}^{j} \exp(-w_i^2/12)$.

In order to check the condition on epoch length in Lemma 10, we notice that with k^* as in (A.2.9), we have

$$c\frac{s}{\gamma}\sqrt{\frac{\log d + \sigma_i^2 w^2}{T_0}} \leq R_1 2^{-k^*/2-1} = \frac{R_{k^*+1}}{2}.$$

Since we assume that constants σ_k are decreasing in k, the inequality also holds for $k \geq k^* + 1$, therefore Lemma 10 applies in this setting.

The setting of epoch length in Theorem 1 ensures that the total number of epochs we perform is

$$k_0 = \log\left(\frac{R_1\gamma}{s}\sqrt{\frac{T}{\log d + \sigma^2 w^2}}\right).$$

Now we have two possibilities. Either $k_0 \leq k^*$ or $k_0 \geq k^*$. In the former, Lemma 9 ensures that the error bound $\|\tilde{\theta}_{k_0} - \theta^*\|_2^2 \leq cR_{k_0}^2/s$. In the latter case, we use Lemma 10 and get the error bound $cR_{k^*}^2/s$. Substituting values of k_0, k^* in these bounds completes the proof.

Proof of Lemma 9 and Lemma 10 follows directly from that of Lemma 5 and Lemma 3 in [Agarwal et al., 2012b].

A.2.7 Proof of Guarantees for Sparse Graphical Model selection Problem

Here we prove Corollary 1. According to C.1, in order to prove guarantees, we first need to bound $\|z_{k+1} - z_k\|_1$ and $\|z_k\|_\infty$. According to Equation (A.2.5) and considering the imposed ℓ_1 bound, this is equivalent to bound $\|g_{k+1} - g_k\|_1$ and $\|g_k\|_\infty$. The rest of the proof follows on lines of Theorem 1 proof. On the other hand, Lipschitz property requires a bound on $\|g_k\|_1$, which is much more stringent.

Assuming we are in a close proximity of Θ^*, we can use Taylor approximation to locally approximate Θ^{-1} by Θ^{*-1} as in [Ravikumar et al., 2011]

$$\Theta^{-1} = \Theta^{*-1} - \Theta^{*-1}\Delta\Theta^{*-1} + \mathcal{R}(\Delta),$$

where $\Delta = \Theta - \Theta^*$ and $\mathcal{R}(\Delta)$ is the remainder term. We have

$$\|g_{k+1} - g_k\|_1 \leq \vert\!\vert\!\vert \Gamma^* \vert\!\vert\!\vert_\infty \|\Theta_{k+1} - \Theta_k\|_1,$$

and

$$\begin{aligned}\|g_k\|_\infty &\leq \|g_k - \mathbb{E}(g_k)\|_\infty + \|\mathbb{E}(g_k)\|_\infty \\ &\leq \|e_k\|_\infty + \|\Sigma^* - \Theta_k^{-1}\|_\infty \leq \sigma + \|\Gamma^*\|_\infty \|\Theta_{k+1} - \Theta_k\|_1.\end{aligned}$$

The term $\|\Theta_{k+1} - \Theta_k\|_1$ is bounded by $2R_i$ by construction. We assume $\vert\!\vert\!\vert \Gamma^* \vert\!\vert\!\vert_\infty$ and $\|\Gamma^*\|_\infty$ are bounded.

The error Δ needs to be "small enough" for the $\mathcal{R}(\Delta)$ to be negligible, and we now provide the conditions for this. By definition, $\mathcal{R}(\Delta) = \sum_{k=2}^{\infty} (-1)^k (\Theta^{*-1}\Delta)^k \Theta^{*-1}$. Using triangle inequality and sub-multiplicative property for Frobenious norm,

$$\|\mathcal{R}(\Delta)\|_{\mathbb{F}} \leq \frac{\|\Theta^{*-1}\|_{\mathbb{F}} \|\Delta\Theta^{*-1}\|_{\mathbb{F}}^2}{1 - \|\Delta\Theta^{*-1}\|_{\mathbb{F}}}.$$

For $\|\Delta\|_{\mathbb{F}} \leq 2R_i \leq \frac{0.5}{\|\Theta^{*-1}\|_{\mathbb{F}}}$, we get

$$\|\mathcal{R}(\Delta)\|_{\mathbb{F}} \leq \|\Theta^{*-1}\|_{\mathbb{F}}.$$

We assume $\|\Sigma^*\|_{\mathbb{F}}$ is bounded.

Note that $\{R_i\}_{i=1}^{k_T}$ is a decreasing sequence and we only need to bound R_1. Therefore, if the variables are closely-related we need to start with a small R_1. For weaker correlations, we can start in a bigger ball. The rest of the proof follows the lines of proof for Theorem 3, by replacing G^2 by $\|\!|\Gamma^*|\!\|_\infty R_i(\sigma + \|\!|\Gamma^*|\!\|_\infty R_i)$. Ignoring the higher order terms gives us Corollary 1.

A.3 Guarantees for REASON 2

First, we provide guarantees for the theoretical case such that epoch length depends on epoch radius. This provides intuition on how the algorithm is designed. The fixed-epoch algorithm is a special case of this general framework. We first state and prove guarantees for general framework. Next, we leverage these results to prove Theorem 1. Let the design parameters be set as

$$T_i \simeq C\left[(s+r+\frac{s+r}{\gamma})^2\left(\frac{\log p+\beta^2(p)\sigma_i^2\log(6/\delta_i)+}{R_i^2}\right)+(s+r+\frac{s+r}{\gamma})\left(\frac{G}{R_i}+\rho_x\right)\right], \tag{A.3.1}$$

$$\begin{aligned}\lambda_i^2 = &\frac{\gamma}{(s+r)\sqrt{T_i}}\sqrt{(R_i^2+\tilde{R}_i^2)\log p+\frac{G^2(R_i^2+\tilde{R}_i^2)}{T_i}+\beta^2(p)(R_i^2+\tilde{R}_i^2)\sigma_i^2\log\frac{3}{\delta_i}}\\ &+\frac{\rho_x(R_i^2+\tilde{R}_i^2)}{T_i}+\frac{\alpha^2}{p^2}+\frac{\beta^2(p)\sigma^2}{T_i}\left(\log p+\log\frac{1}{\delta_i}\right),\end{aligned}$$

$$\mu_i^2 = c_\mu\lambda_i^2, \quad \rho \propto \sqrt{\frac{T_i\log p}{R_i^2+\tilde{R}_i^2}}, \quad \rho_x > 0, \quad \tau = \rho.$$

Theorem 11. *Under assumptions $A2 - A6$ and parameter settings as in* (A.3.1), *there exists a constant $c_0 > 0$ such that REASON 2 satisfies the following for all $T > k_T$,*

$$\begin{aligned}&\|\bar{S}(T)-S^*\|_{\mathbb{F}}^2+\|\bar{L}(T)-L^*\|_{\mathbb{F}}^2 \leq\\ &\frac{c_0(s+r)}{T}\left[\log p+\beta^2(p)\sigma^2\left(w^2+\log k_T\right)\right]+\left(1+\frac{s+r}{\gamma^2 p}\right)\frac{\alpha^2}{p}.\end{aligned}$$

with probability at least $1-6\exp(-w^2/12)$ and

$$k_T \simeq -\log\left(\frac{(s+r)^2}{\gamma^2R_1^2T}\left[\log p+\beta^2(p)\sigma^2w^2\right]\right).$$

For Proof outline and detailed proof of Theorem 11 see Appendix A.3.1 and A.4 respectively.

A.3.1 Proof outline for Theorem 11

The foundation block for this proof is Proposition 3.

Proposition 3. *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* (A.3.1). *Then, there exists a universal constant c such that for any radius $R_i, \tilde{R}_i$, $\tilde{R}_i = c_r R_i, 0 \leq c_r \leq 1$,*

$$f(\bar{M}(T_i)) + \lambda_i \phi(\bar{W}(T_i)) - f(\hat{M}_i) - \lambda_i \phi(\hat{W}(T_i)) \tag{A.3.2a}$$

$$\begin{aligned}
&\leq \sqrt{\frac{R_i^2 + \tilde{R}_i^2}{T_i}} \sqrt{\log p} + \frac{R_i^2 + \tilde{R}_i^2}{T_i} \rho_x + \frac{G\sqrt{R_i^2 + \tilde{R}_i^2}}{T_i} \\
&\quad + \frac{\beta(p)(R_i + \tilde{R}_i)\sigma_i \sqrt{12 \log \frac{3}{\delta_i}}}{\sqrt{T_i}} + \frac{\beta(p) G (R_i + \tilde{R}_i)\sigma_i \sqrt{12 \log \frac{3}{\delta_i}}}{T_i \sqrt{\log p}},
\end{aligned}$$

$$\|\bar{S}(T_i) - S^*\|_1^2 \leq \frac{c'}{\sqrt{C}} R_i^2 + c(s + r + \frac{(s+r)^2}{p\gamma^2}) \frac{\alpha^2}{p}, \tag{A.3.2b}$$

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

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

In order to prove Proposition 3, 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 12. *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 13. *Under the conditions of Proposition 3 and with parameter settings* (A.3.1), (A.3.1), *we have*

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

with probability at least $1 - \delta_i$.

A.4 Proof of Theorem 11

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 3 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 (A.3.1) where $C \geq 1$ is a constant we can choose. By substituting this T_i in inequality (A.3.2b), the simplified bound (A.3.2b) 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 12 and 13 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 $w_i^2 = w^2 + 24\log i$, 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 3, we get

$$k_T \simeq -\log \frac{(s+r+(s+r)/\gamma)^2}{R_1^2 T} - \log\left[\log p + 12\beta^2(p)\sigma^2 w^2\right].$$

As a result

$$\|\Delta_S^*(T_i)\|_{\mathbb{F}}^2 \leq \frac{C(s+r)}{T}\left[\log p + \beta^2(p)\sigma^2\left(w^2 + \log k_T)\right]\right] + \frac{\alpha^2}{p}.$$

For the low-rank part, we proved feasibility in proof of Equation (A.3.2b), consequently The same bound holds for $\|\Delta_L^*(T_i)\|_{\mathbb{F}}^2$.

A.4.1 Proofs for Convergence within a Single Epoch for Algorithm 3

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. With

$\rho = c_2\sqrt{T_i}, \tau = \rho, c_2 = \frac{\sqrt{\log p}}{\sqrt{R_i^2+\tilde{R}_i^2}}$. We use the same approach as in Section A.2.1 for bounds on dual variable Z_k. Hence,

$$
\begin{aligned}
&f(\bar{M}(T_i)) + \lambda_i\phi(\bar{W}(T_i)) - f(\hat{M}_i) - \lambda_i\phi(\hat{W}(T_i)) \\
&\leq \frac{c_2\|A\hat{W}(T_i) - AW_0\|_{\mathbb{F}}^2}{\sqrt{T_i}} + \frac{\rho_x\|\hat{M}(T_i) - M_0\|_{\mathbb{F}}^2}{T_i} + \frac{GR_i}{T_i} + \frac{R_i\sqrt{\log p}}{\sqrt{T_i}} \\
&\quad + \frac{\sum_{k=1}^{T_i}\operatorname{Tr}(E_k, \hat{M}_i - M_k)}{T_i} \\
&\leq \left[\frac{c_2}{\sqrt{T_i}} + \frac{\rho_x}{T_i}\right]\|\hat{S}_i - \tilde{S}_i + \hat{L}_i - \tilde{L}_i\|_{\mathbb{F}}^2 + \frac{GR_i}{T_i} + \frac{R_i\sqrt{\log p}}{\sqrt{T_i}} \\
&\quad + \frac{\sum_{k=1}^{T_i}\operatorname{Tr}(E_k, \hat{M}_i - M_k)}{T_i}.
\end{aligned}
$$

By the constraints enforced in the algorithm, we have

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

Lemma 14. *The dual variable in REASON 2 is bounded. i.e.,*

$$\|Z_k\|_1 \leq G + 2\rho_0 R_i, \quad \textit{where} \;\; \rho_0 := \rho_x + \rho.$$

Proof. The proof follows the same line as in proof of Lemma 8 and replacing θ, y by M, W where $W = [S; L]$. Hence,

$$\|Z_k\|_1 \leq G + 2\rho_0 R_i, \quad \text{where} \quad \rho_0 := \rho_x + \rho.$$

■

A.4.2 Proof of Proposition 3: Equation (A.3.2a)

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

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

Hence,

$$\begin{aligned}
&[\mathrm{Tr}(E_k, \hat{M}_i - M_k)]^2 \\
&\leq [\|E_k\|_\infty \|S_k - \hat{S}_i\|_1 + \|E_k\|_2^2 \|L_k - \hat{L}_i\|_* + \|E_k\|_\infty \|(Z_{k+1} - Z_k)/\tau\|_1]^2 \\
&\leq [2R_i \|E_k\|_\infty + 2\tilde{R}_i \|E_k\|_2 + (G + 2\rho_0 R_i)/\tau \|E_k\|_\infty]^2 \\
&\leq \|E_k\|_2^2 [2R_i + 2\tilde{R}_i + (G + 2\rho_0 R_i)/\tau]^2.
\end{aligned}$$

Consider the term $\|E_k\|_2$. Using Assumption A4, our previous approach in proof of Equation (A.1.3a), holds true with addition of a $\beta(p)$ term. Consequently,

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

with probability at least $1 - \delta_i$.

A.4.3 Proof of Lemma 12

We use Lemma 1 [Negahban et al., 2012] for designing λ_i and μ_i. This Lemma requires that for optimization problem $\min_{\Theta}\{L(\Theta)+\lambda_i Q(\Theta)\}$, we design the regularizer coefficient $\lambda_i \geq 2Q^*(\nabla L(\Theta^*))$, where L is the loss function, Q is the regularizer and Q^* is the dual regularizer. For our case Θ stands for $[S; L]$.

$$L(\Theta) = \frac{1}{n} \sum_{k=1}^{n} f_k(\Theta, x),$$

and

$$\begin{aligned}
Q^*(\nabla L(\Theta^*)) &= Q^* \left[\mathbb{E}(\nabla f(\Theta^*) + \frac{1}{n} \sum_{k=1}^{n} \{\nabla f_k(\Theta^*)) - \mathbb{E}(\nabla f(\Theta^*))\} \right] \\
&= Q^*(\frac{1}{n} \sum_{k=1}^{n} E_k),
\end{aligned}$$

where $E_k = g_k - \mathbb{E}(g_k)$ is the error in gradient estimation as defined earlier.

Using Theorem 1 [Agarwal et al., 2012a] in this case, if we design

$$\lambda_i \geq 4 \left\| \frac{1}{n} \sum_{k=1}^{n} E_k \right\|_\infty + \frac{4\gamma\alpha}{p} \quad \text{and} \quad \mu_i \geq 4 \left\| \frac{1}{n} \sum_{k=1}^{n} E_k \right\|_2, \tag{A.4.1}$$

then we have

$$\|\hat{S}_i - S^*\|_{\mathbb{F}}^2 + \|\hat{L}_i - L^*\|_{\mathbb{F}}^2 \leq c\{s\frac{\lambda_i^2}{\gamma^2} + r\frac{\mu_i^2}{\gamma^2}\}. \tag{A.4.2}$$

Lemma 15. *Assume $X \in \mathbb{R}^{p\times p}$. If $\|X\|_2 \leq B$ almost surely then with probability at least $1-\delta$ we have*

$$\left\| \frac{1}{n} \sum_{k=1}^{n} X_k - \mathbb{E}(X_k) \right\|_2 \leq \frac{6B}{\sqrt{n}} \left(\sqrt{\log p} + \sqrt{\log \frac{1}{\delta}} \right).$$

Note that this lemma is matrix Hoeffding bound and provides a loose bound on matrix. Whereas using matrix Bernstein provided tighter results using $\mathbb{E}(E_k E_k^\top)$. Moreover, since the elementwise max norm $\|X\|_\infty \leq \|X\|_2$, we use the same upper bound for both norms.

By definition $\mathbb{E}(E_k) = 0$. According to Assumption A4, $\|E_k\|_2 \leq \beta(p)\sigma$. Thus it suffices to design

$$\lambda_i \geq \frac{24\beta(p)\sigma_i}{\sqrt{T_i}}\left(\sqrt{\log p} + \sqrt{\log\frac{1}{\delta_i}}\right) + \frac{4\gamma\alpha}{p}$$

and

$$\mu_i \geq \frac{24\beta(p)\sigma_i}{\sqrt{T_i}}\left(\sqrt{\log p} + \sqrt{\log\frac{1}{\delta_i}}\right).$$

Then, we can use Equation (A.4.2).

A.4.4 Proof of Lemma 13

By LSC condition on $X = S + L$

$$\begin{aligned}
&\frac{\gamma}{2}\|\hat{S}_i - \bar{S}(T_i) + \hat{L}_i - \bar{L}(T_i)\|_{\mathbb{F}}^2 \\
&\leq f(\bar{X}(T_i)) + \lambda_i\|\bar{S}(T_i)\|_1 + \mu_i\|\bar{L}(T_i)\|_* - f(\hat{X}_i) - \lambda_i\|\hat{S}(T_i)\|_1 - \mu_i\|\hat{L}(T_i)\|_*
\end{aligned}$$

We want to use the following upper bound for the above term.

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

$\hat{M}_i = \hat{X}_i$, i.e., all the terms are the same except for $f(\bar{M}(T_i)), f(\bar{X}(T_i))$. We have $\bar{M}(T_i) = \bar{X}(T_i) + \frac{Z_T}{\tau T_i}$. This is a bounded and small term $\mathcal{O}(R_i/(T_i\sqrt{T_i}))$. We accept this approximation giving the fact that this is a higher order term compared to $\mathcal{O}(1/\sqrt{T_i})$. Hence, it will not play a role in the final bound on the convergence rate. Therefore,

$$\begin{aligned} &\frac{\gamma}{2}\|\hat{S}_i - \bar{S}(T_i) + \hat{L}_i - \bar{L}(T_i)\|_{\mathbb{F}}^2 \qquad \text{(A.4.3)} \\ &\leq \sqrt{\frac{R_i^2 + \tilde{R}_i^2}{T_i}}\sqrt{\log p} + \frac{R_i^2 + \tilde{R}_i^2}{T_i}\rho_x + \frac{G\sqrt{R_i^2 + \tilde{R}_i^2}}{T_i} \\ &\quad + \frac{\beta(p)(R_i + \tilde{R}_i)\sigma_i\sqrt{12\log\frac{3}{\delta_i}}}{\sqrt{T_i}} + \frac{\beta(p)G(R_i + \tilde{R}_i)\sigma_i\sqrt{12\log\frac{3}{\delta_i}}}{T_i\sqrt{\log p}}, \end{aligned}$$

with probability at least $1 - \delta_i$.

For simplicity, we use

$$\begin{aligned} H_1 = &\sqrt{\frac{R_i^2 + \tilde{R}_i^2}{T_i}}\sqrt{\log p} + \frac{R_i^2 + \tilde{R}_i^2}{T_i}\rho_x + \frac{G\sqrt{R_i^2 + \tilde{R}_i^2}}{T_i} \\ &+ \frac{\beta(p)(R_i + \tilde{R}_i)\sigma_i\sqrt{12\log\frac{3}{\delta_i}}}{\sqrt{T_i}} + \frac{\beta(p)G(R_i + \tilde{R}_i)\sigma_i\sqrt{12\log\frac{3}{\delta_i}}}{T_i\sqrt{\log p}}. \end{aligned}$$

We have,

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

In addition,

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

We have,

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

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

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

$$\|U_{k+1}\|_\infty = \|\nabla\{\|L\|_*\}\|_\infty \leq \sqrt{\operatorname{rank}(L)} \leq \sqrt{p}.$$

Last step is from the analysis of Watson [1992]. Therefore,

$$\gamma \| \operatorname{Tr}(\hat{\Delta}_S(T_i)\hat{\Delta}_L(T_i))| \leq \gamma(\frac{2\alpha}{p} + \frac{\sqrt{p}}{\tau T_i})\|\hat{\Delta}_S(T_i)\|_1.$$

Consequently,

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

Combining the above equation with (A.4.3), we get

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

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

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

with probability at least $1 - \delta_i$.

A.4.5 Proof of Proposition 3: Equation (A.3.2b)

Now we want to convert the error bound in (A.3.2a) from function values into vectorized ℓ_1 and Frobenius-norm bounds. Since the error bound in (A.3.2a) holds for the minimizer $\hat{M}_i$, it also holds for any other feasible matrix. In particular, applying it to M^* leads to,

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

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

$$\phi(\bar{W}(T_i)) - \phi(W^*) \leq H_1/\lambda_i. \tag{A.4.4}$$

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,

$$\|(\bar{S}(T_i))_{supp}\|_1 \geq \|(S^*)_{supp}\|_1 - \|(\Delta_S^*)_{supp}\|_1$$
$$\|(\bar{S}(T_i))_{supp^c}\|_1 \geq -\|(S^*)_{supp^c}\|_1 + \|(\Delta_S^*)_{supp^c}\|_1,$$

and

$$\|(\bar{L}(T_i))_V\|_* \geq \|(L^*)_V\|_* - \|(\Delta_L^*)_V\|_*$$
$$\|(\bar{L}(T_i))_{V^\perp}\|_* \geq -\|(L^*)_{V^\perp}\|_* + \|(\Delta_L^*)_{V^\perp}\|_*.$$

Consequently,

$$\|(\Delta_S^*)_{supp^c}\|_1 + \frac{\mu_i}{\lambda_i}\|(\Delta_L^*)_{V^\perp}\|_* \leq \|(\Delta_S^*)_{supp}\|_1 + \frac{\mu_i}{\lambda_i}\|(\Delta_L^*)_V\|_* + H_1/\lambda_i. \qquad \text{(A.4.5)}$$

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

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

Hence,

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

As Equation (A.4.1) is satisfied, we can use Lemma 1 [Negahban et al., 2012]. Combining the result with Lemma 12, 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 12 and the inequality $\|(\hat{\Delta}_S(T_i))_{supp}\|_1 \leq \sqrt{s}\|\hat{\Delta}(T_i)\|_{\mathbb{F}}$ allows us to conclude that there exists a universal constant c such that

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

with probability at least $1-\delta_i$. Optimizing the above bound with choice of λ_i and complying with the conditions in Lemma 15, leads to

$$\lambda_i^2 = \frac{\gamma}{s+r}H_1 + \frac{\alpha^2}{p^2} + \frac{\beta^2(p)\sigma^2}{T_i}\left(\log p + \log\frac{1}{\delta}\right).$$

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

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

we have

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

Therefore,

$$\begin{aligned}\|\Delta_S^*(T_i)\|_1^2 &\le 2\|\hat{\Delta}_S(T_i)\|_1^2 + 2\|S^* - \hat{S}_i\|_1^2 \\ &\le 2\|\hat{\Delta}(T_i)\|_1^2 + 8c(s+r)(s\frac{\lambda_i^2}{\gamma^2}+r\frac{\mu_i^2}{\gamma^2}) \\ &\le c(s+r+\frac{s+r}{\gamma})H_1 + c(s+r)(1+\frac{s+r}{p\gamma^2})\frac{\alpha^2}{p} + (s+r)(\frac{p^2}{\tau T_i^2}+\frac{\alpha}{\tau T_i}).\end{aligned} \tag{A.4.6}$$

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 $0 \le c_r \le 1$. 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 (A.4.6). 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 (A.4.6) is not greater that $R_i^2/2$. All above conditions are met if we choose the epoch length

$$\begin{aligned} T_i \simeq\, & C(s+r+\frac{s+r}{\gamma})^2 \left[\frac{\log p + 12\beta^2(p)\sigma_i^2 \log\frac{6}{\delta}}{R_i^2}\right] \\ & + C(s+r+\frac{s+r}{\gamma}) \left[\frac{\beta(p)G\sigma_i\sqrt{12\log\frac{6}{\delta}}}{R_i\sqrt{\log p}} + \frac{G}{R_i} + \rho_x\right], \end{aligned}$$

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

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

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

With similar analysis for L, we get

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

A.4.6 Proof of Guarantees with Fixed Epoch Length, Sparse + Low Rank Case

This is a special case of Theorem 11 (Appendix). Note that this fixed epoch length results in a convergence rate that is worse by a factor of $\log p$. The key difference between this case and optimal epoch length setting of Theorem 11 is that in the latter we guaranteed error halving by the end of each epoch whereas with fixed epoch length that statement may not be possible after the number of epochs becomes large enough. Therefore, we need to show that in such case the error does not increase much to invalidate our analysis. Let k^* be the epoch number such that error halving holds true until then. Next we demonstrate that error does not increase much for $k > k^*$. The proof follows the same nature as that of Theorem 1 (in the main text), Section A.2.6, with

$$k^* := \sup \left\{ i : 2^{\frac{j}{2}+1} \leq \frac{cR_1\gamma}{s+r}\sqrt{\frac{T_0}{\log p + \beta^2(p)\sigma_i^2 w^2}} \right\},$$

for all epochs $j \leq i$ and

$$k_0 = \log\left(\frac{R_1\gamma}{s+r}\sqrt{\frac{T}{\log p + \beta^2(p)\sigma^2 w^2}}\right).$$

A.4.7 Proof of Guarantees for Sparse + Low Rank Graphical Model selection Problem

Here we prove Corollary 2. Proof follows by using the bounds derived in Appendix A.2.7 for Taylor series expansion and following the lines of Theorem 11 proof as in Appendix A.4.

According to D.1, in order to prove guarantees, we first need to bound $\|z_{k+1} - z_k\|_1$ and $\|z_k\|_\infty$. According to Equation (A.2.5) and considering the imposed ℓ_1 bound, this is equivalent to bound $\|g_{k+1} - g_k\|_1$ and $\|g_k\|_\infty$.$\|g_{k+1} - g_k\|_1$ and $\|g_k\|_\infty$. The rest of the proof follows on lines of Theorem 2 proof. On the other hand, Lipschitz property requires a bound on $\|g_k\|_1$, which is much more stringent.

Assuming we are in a close proximity of M^*, we can use Taylor approximation to locally approximate M^{-1} by M^{*-1} as in [Ravikumar et al., 2011]

$$M^{-1} = M^{*-1} - M^{*-1}\Delta M^{*-1} + \mathcal{R}(\Delta),$$

where $\Delta = M - M^*$ and $\mathcal{R}(\Delta)$ is the remainder term. We have

$$\|g_{k+1} - g_k\|_1 \leq |||\Gamma^*|||_\infty \|M_{k+1} - M_k\|_1,$$

and

$$\begin{aligned}\|g_k\|_\infty &\leq \|g_k - \mathbb{E}(g_k)\|_\infty + \|\mathbb{E}(g_k)\|_\infty \\ &\leq \|e_k\|_\infty + \|\Sigma^* - M_k^{-1}\|_\infty \\ &\leq \sigma + \|\Gamma^*\|_\infty \|M_{k+1} - M_k\|_1.\end{aligned}$$

The term $\|M_{k+1} - M_k\|_1$ is bounded by $2\breve{R}$ by construction. We assume $|\!|\!|\Gamma^*|\!|\!|_\infty$ and $\|\Gamma^*\|_\infty$ are bounded.

The error Δ needs to be "small enough" for the $\mathcal{R}(\Delta)$ to be negligible, and we now provide the conditions for this. By definition, $\mathcal{R}(\Delta) = \sum_{k=2}^\infty (-1)^k ({M^*}^{-1}\Delta)^k {M^*}^{-1}$. Using triangle inequality and sub-multiplicative property for Frobenious norm,

$$\|\mathcal{R}(\Delta)\|_\mathbb{F} \leq \frac{\|{M^*}^{-1}\|_\mathbb{F} \|\Delta {M^*}^{-1}\|_\mathbb{F}^2}{1 - \|\Delta {M^*}^{-1}\|_\mathbb{F}}.$$

For $\|\Delta\|_\mathbb{F} \leq 2\breve{R} \leq \frac{0.5}{\|{M^*}^{-1}\|_\mathbb{F}}$, we get

$$\|\mathcal{R}(\Delta)\|_\mathbb{F} \leq \|{M^*}^{-1}\|_\mathbb{F}.$$

We assume $\|\Sigma^*\|_\mathbb{F}$ is bounded.

Therefore, if the variables are closely-related we need to start with a small $\breve{R}$. For weaker correlations, we can start in a bigger ball. The rest of the proof follows the lines of proof for Theorem 11, by replacing G^2 by $|\!|\!|\Gamma^*|\!|\!|_\infty \breve{R}(\sigma + \|\Gamma^*\|_\infty \breve{R})$.

Bibliography

Pictorial explanation of a border node. August 2014. URL https://www.dropbox.com/s/jc4jo4t26ma2mlx/border.jpg?dl=0.

A. Abur and A.G Exposito. *Power System State Estimation, Theory and Implementation*. Marcel Dekker, 2004.

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.

S Massoud Amin and Anthony M Giacomoni. Smart grid-safe, secure, self-healing. *IEEE Power Energy Mag*, 10:33–40, 2012.

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.

Emilio Ancillotti, Raffaele Bruno, and Marco Conti. The role of communication systems in smart grids: Architectures, technical solutions and research challenges. *Computer Communications*, 36(17):1665–1697, 2013.

B. De Finetti. *Theory of Probability*. Wiley, 1975.

R. Banirazi and E. Jonckheere. Geometry of power flow in negatively curved power grids: Toward a smart transmission system. In *IEEE CDC*, pages 6259–6264, 2010.

C. M. Bishop. *Pattern recognition and machine learning*. Springer, New York, USA, 2006.

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.

R.S Blum C. Wei, A. Wiesel. Change detection in smart grids using errors in variables models. In *Sensor Array and Multichannel Signal Processing Workshop (SAM), 2012 IEEE 7th*, pages 16–20, June 17-20 2012.

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.

R. Diao, K. Sun, V. Vittal, R. OKeefe, M. Richardson, N. Bhatt, D. Stradford, and S. Sarawgi. Decision tree-based online voltage security assesment using PMU measurements. *IEEE Trans. Power Systems*, 24:832–839, May 2009.

JF Dopazo, OA Klitin, and AM Sasson. Stochastic load flows. *Power Apparatus and Systems, IEEE Trans.*, 94(2):299–309, 1975.

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.

J. Friedman, T. Hastie, and R. Tibshirani. Sparse inverse covariance estimation with the graphical Lasso. *Biostatistics*, 2007.

A. Giani, E. Bitar, M. Garcia, M. McQueen, P. Khargonekar, and K. Poolla. Smart grid data integrity attacks. *IEEE Trans. Smart Grid*, 1(1), January 2012.

T Goldstein, B. ODonoghue, and S. Setzer. Fast alternating direction optimization methods. *CAM report*, pages 12–35, 2012.

M. He and J. Zhang. A dependency graph approach for fault detection and localization towards secure smart grid. *IEEE Trans. Smart Grid*, 2:342–351, June 2011.

Julien M Hendrickx, Karl Henrik Johansson, Raphaël M Jungers, Henrik Sandberg, and Kin Cheong Sou. Efficient computations of a security index for false data attacks in power networks. *IEEE TRANSACTIONS ON AUTOMATIC CONTROL*, 59(12), 2014.

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.

N. Liu Ide T. Lozano, A.C. Abe. Proximity-based anomaly detection using sparse structure learning. In *SIAM International Conference on Data Mining*, Philadelphia, 2009.

M. Janzamin and A. Anandkumar. High-Dimensional covariance decomposition into sparse Markov and independence models. *JMLR*, 15:1549–1591, April 2014.

Majid Janzamin and Animashree Anandkumar. High-dimensional covariance decomposition into sparse markov and independence domains. In *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*, pages 1839–1846, 2012.

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.

D.R. Anderson K. P. Burnham. Model Selection and Multi-Model Inference: A Practical Information-Theoretic Approach. page 51, 2002.

Ashwin Kashyap and Duncan Callaway. Estimating the probability of load curtailment in power systems with responsive distributed storage. In *IEEE PMAPS*, pages 18–23, 2010.

Oliver Kosut, Liyan Jia, Robert J Thomas, and Lang Tong. Malicious data attacks on smart grid state estimation: Attack strategies and countermeasures. In *IEEE SmartGridComm, 2010*, pages 220–225, 2010.

R.A. Kullback, S.; Leibler. On Information and Sufficiency. *Annals of Mathematical Statistics*, (22(1)):79–86, 1951.

R.A. Kullback, S.; Leibler. Letter to the Editor: The KullbackLeibler distance. *The American Statistician*, (41(4)):340341, 1987.

S Kullback. *Information theory and statistics*. John Wiley and sons, NY, 1959.

Cheolhyeon Kwon, Weiyi Liu, and Inseok Hwang. Security analysis for cyber-physical systems against stealthy deception attacks. In *American Control Conference (ACC), 2013*, pages 3344–3349. IEEE, 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.

Z. Lin, M. Chen, and Y. Ma. The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. *arXiv preprint arXiv:1009.5055*, 2010.

J. Lofberg. YALMIP: A toolbox for modeling and optimization in MATLAB. In *IEEE CACSD*, September 2004. Available from http://users.isy.liu.se/johanl/yalmip/.

Po-Ling Loh and Martin J Wainwright. Regularized m-estimators with nonconvexity: Statistical and algorithmic theory for local optima. In *Advances in Neural Information Processing Systems*, pages 476–484, 2013.

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.

J. Mur-Amada and J. Salln-Arasanz. From turbine to wind farms - technical requirements and spin-off products. In Gesche Krause, editor, *Phase Transitions and Critical Phenomena*, volume 18, pages 101–132. InTech, April 2011.

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.

Guodong Pang, George Kesidis, and Takis Konstantopoulos. Avoiding overages by deferred aggregate demand for pev charging on the smart grid. In *Communications (ICC), 2012 IEEE International Conference on*, pages 3322–3327. IEEE, 2012.

Jim Pitman and Nathan Ross. Archimedes, gauss, and stein. *Notices AMS*, 59: 1416–1421, 2012.

S. R. Rajagopalan, L. Sankar, S. Mohajer, and H. V. Poor. Smart meter privacy: A utility-privacy framework. In *2nd Annual IEEE Conference on Smart Grid Communications*, Brussels, Belgium, Oct 2011.

G. Raskutti, M. J. Wainwright, and B. Yu. Minimax rates of estimation for high-dimensional linear regression over ℓ_q-balls. *IEEE Trans. Information Theory*, 57 (10):6976—6994, October 2011.

Pradeep Ravikumar, Martin J Wainwright, Garvesh Raskutti, Bin Yu, et al. High-dimensional covariance estimation by minimizing 1-penalized log-determinant divergence. *Electronic Journal of Statistics*, 5:935–980, 2011.

Nicolas Le Roux, Mark Schmidt, and Francis Bach. A stochastic gradient method with an exponential convergence rate for strongly-convex optimization with finite training sets. Technical report, 2012.

A Schellenberg, W. Rosehart, and J. Aguado. Cumulant-based probabilistic optimal power flow with Gaussian and Gamma distributions. *Power Systems, IEEE Trans.*, 20(2):773–781, 2005.

Hanie Sedghi and Edmond Jonckheere. Statistical structure learning of smart grid for detection of false data injection. In *Power and Energy Society General Meeting (PES), 2013 IEEE*, pages 1–5. IEEE, 2013.

Hanie Sedghi and Edmond Jonckheere. On conditional mutual information in Gauss-Markov structured grids. In G. Como, B. Bernhardson, and A. Rantzer, editors, *Lecture notes in Control and Information Sciences*, volume 450, pages 277–297. Springer, 2014.

Hanie Sedghi and Edmond Jonckheere. Statistical structure learning to ensure data integrity in smart grid. *Smart Grid, IEEE Transactions on*, 6, 2015.

Hanie Sedghi, Anima Anandkumar, and Edmond Jonckheere. Guarantees for multi-step stochastic admm in high dimensions. *arXiv preprint arXiv:1402.5131*, 2014a. submitted to Journal of Machine Learning Research(JMLR).

Hanie Sedghi, Anima Anandkumar, and Edmond Jonckheere. Multi-step stochastic admm in high dimensions: Applications to sparse optimization and matrix decomposition. In *Proceedings of the 28th Annual Conference on Neural Information Processing Systems (NIPS-14)*, pages 2771–2779, 2014b.

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.

André Teixeira, György Dán, Henrik Sandberg, and Karl Henrik Johansson. A cyber security study of a scada energy management system: Stealthy deception attacks on the state estimator. In *World Congress*, volume 18, pages 11271–11277, 2011.

K. C. Toh, M.J. Todd, and R. H. Tutuncu. SDPT3 - a MATLAB software package for semidefinite programming. *Optimization Methods and Software*, 11:545–581, 1999.

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.

Zhaoran Wang, Han Liu, and Tong Zhang. Optimal computational and statistical rates of convergence for sparse nonconvex learning problems. *arXiv preprint arXiv:1306.4960*, 2013c.

G. Watson. Characterization of the subdifferential of some matrix norms. *Linear Algebra and its Applications*, 170(0):33–45, 1992.

Peng Ning Yao Liu and Michael K. Reiter. False data injection attacks against state estimation in electric power grids. *ACM trans. Information and Security Systems*, 14, May 2011.

Pei Zhang and Stephen T Lee. Probabilistic load flow computation using the method of combined cumulants and gram-charlier expansion. *Power Systems, IEEE Tran.*, 19(1):676–682, 2004.

H. Zhu and G. B. Giannakis. Sparse overcomplete representations for efficient identification of power line outages. IEEE Tran. on Power Systems, 2012.

Kun Zhu, Moustafa Chenine, Lars Nordström, Sture Holmström, and Göran Ericsson. An empirical study of synchrophasor communication delay in a utility TCP/IP network. *International Journal of Emerging Electric Power Systems*, 14 (4):341–350, 2013.

R. D. Zimmerman, C. E. Murillo-Snchez, and R. J. Thomas. Matpower steady-state operations, planning and analysis tools for power systems research and education. *Power Systems, IEEE Transactions on*, 26(1):12–19, Feb. 2011.