Optimization Methods for Machine Learning under Structural Constraints

by

Wenyu Chen

B.S., Peking University (2016)
M.S., Columbia University in the City of New York (2018)

Submitted to the Sloan School of Management
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Operations Research

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2023

Author .....
Sloan School of Management
May 2, 2023

Certified by .....
Rahul Mazumder
Associate Professor of Operations Research and Statistics
Thesis Supervisor

Accepted by .....
Patrick Jaillet
Dugald C. Jackson Professor, Department of Electrical Engineering and Computer Science
Co-Director, Operations Research Center

Optimization Methods for Machine Learning under Structural Constraints

by

Wenyu Chen

Submitted to the Sloan School of Management
on May 2, 2023, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Operations Research

Abstract

In modern statistical and machine learning models, structural constraints are usually imposed for model interpretability as well as model complexity reduction. In this thesis, we present scalable optimization methods for several large-scale machine learning problems under structural constraints, with a focus on shape constraints in nonparametric statistics and sparsity in high-dimensional statistics.

In the first chapter, we consider the subgradient regularized convex regression problem, which aims to fit a convex function between the target variable and covariates. We propose novel large-scale algorithms, based on proximal gradient descent and active set methods, and derive novel linear convergence guarantees for our proposed algorithms. Empirically, our framework can approximately solve instances with $n = 10^5$ and $d = 10$ within minutes.

In the second chapter, we develop a new computational framework for computing log-concave density MLE, based on smoothing techniques in combination with an appropriate integral discretization of increasing accuracy. We establish convergence guarantees of our approaches and demonstrate significant runtime improvements over earlier convex approaches.

In the third chapter, we focus on Gaussian Graphical Models, which aims to estimate a sparse precision matrix from iid multivariate Gaussian samples. We propose a novel estimator via $\ell_0\ell_2$-penalized pseudolikelihood. We then design a specialized nonlinear Branch-and-Bound (BnB) framework that solves a mixed integer programming (MIP) formulation of the proposed estimator. Our estimator is computationally scalable to $p \sim 10{,}000$, and provides faster runtime compared to competing ℓ_1 approaches, while leading to superior statistical performance.

In the fourth chapter, we further look into improving the BnB framework for sparse learning problems with the $\ell_0\ell_2$ penalty and general convex smooth losses. We

present a novel screening procedure within the BnB framework to fix relaxed variables to 0 or 1 with guarantees. Our experiments indicate that this screening procedure can significantly reduce the runtimes of BnB solvers.

Thesis Supervisor: Rahul Mazumder
Title: Associate Professor of Operations Research and Statistics

Acknowledgments

First and foremost, I would like to express my sincere gratitude to my advisor Rahul Mazumder for his unwavering support, encouragement, and contagious enthusiasm throughout my PhD program. His passion for our research area has fueled my own interest and inspired me to think creatively. We have spent a lot of time and had countless meetings to discuss research ideas and think about challenging problems. He taught me so much about how to conduct research, how to think about problems at a high level, and how to push myself to do my best work. His guidance and mentorship have been invaluable in shaping my research ideas and methodology, and I feel incredibly fortunate to have had such a dedicated and enthusiastic advisor.

Next, I extend my sincere thanks to my esteemed thesis committee members, Rahul Mazumder, Andy Sun, and Bart Van Parys, for their invaluable support and insightful advice. I appreciate their dedication and commitment to ensuring the quality of my dissertation. Additionally, I would like to express my gratitude to Bart Van Parys and Sasha Rakhlin for serving on my general exam committee and providing valuable feedback and comments.

I am deeply grateful to all my colleagues at MIT who have contributed to my research journey. In particular, I would like to thank Hussein for his valuable advice on research and PhD life, Shibal Ibrahim for his invaluable expertise in deep learning, Kayhan Behdin for his critical insights into statistics, and Haoyue Wang, Riade Benbaki, and Xiang Meng for their extensive support in various research projects. I am truly grateful to all of them for being wonderful colleagues and friends. Also, I want to acknowledge Danying Xiao, Xiaming Jin, Brian Hsu, Peijun Xu, Tim Nonet, Mathieu Sibue, Max R. Tell, Paul Theron, Muzhi Ma, Zhehan Xu and Hengyu Fu for their help in multiple projects. Their contributions have been essential to our success.

I have been very fortunate to interact with many faculty members and researchers outside MIT. I would like to express my profound appreciation for Professor Richard J. Samworth. He has taught me a great deal about conducting rigorous research, provided insights into statistics, and offered constructive feedback that has helped

refine and improve our work. I also want to extend my gratitude to our collaborators Yada Zhu, Yang Zhang, and Pin-Yu Chen at MIT-IBM Lab for their guidance and support, and especially Yada Zhu, who mentored me at IBM and taught me how to work on real-world financial problems. Finally, I want to thank our collaborators at Google, Natalia Ponomareva and Zhe Zhao, for their valuable insights and suggestions on deep learning research.

I am indebted to many people whom I met before my PhD. Professor Donald Goldfarb, my research advisor at Columbia IEOR, deserves special recognition for introducing me to the field of optimization and operations research and supporting my growth as a researcher. I am also grateful to Martin Haugh for his valuable guidance during my decision to pursue a PhD. In addition, I would like to thank Krzysztof Choromanski, Garud Iyengar, Agostino Capponi, Jenny Mak, Chaoxu Zhou, Robin Tang, and Wenjun Wang for their support during my master's study, as well as Jingping Yang, Yangbo He, and Chenxu Li from Peking University for their mentorship.

I would like to extend my appreciation to the many faculty and staff members at MIT who have helped me along the way, including Dimitris Bertsimas, Georgia Perakis, Patrick Jaillet, Robert Freund, Juan Pablo Vielma, Daniel Freund, Laura Rose, and Andrew Carvalho.

I would like to express my gratitude to my friends: Yuchen, Jason, Irra, Qingyang, Michael, Renbo, Xiaoqi, Buxuan, Zhechao, Zihao, Jing, Sean, Li, Xiaoyue, Yu, Rainy, Zikai, Qinyi, Kamessi, Yifei, Jiyi, Zhirong, Xiaohong, Yihao, Yongyi, Yao, Qi, Dorothy, Amber, Siyue, Lubing, Weixin, Guanpeng, Yujia, among many others.

Lastly, I want to thank my family – my father Zhending Chen, my mother Haiping Wu, as well as my late grandfather Yihe Wu and my late aunt Hongying Chen – for their endless and unconditional love and support. I dedicate this thesis to them.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Last few decades have seen substantial advancements in machine learning models, motivated by many applications from various domains. In modern statistical and machine learning models, structural constraints are usually imposed to (i) meet complicated application-specific assumptions and provide model interpretability, and (ii) reduce the sample complexity and improve model performance. While there have been extensive studies in statistical properties of these models, the computational inscalability of underlying optimization problems becomes the main bottlenock for their applications in practice. In this thesis, we present efficient optimization methods for large-scale machine learning problems under structural constraints, with a focus on two types of structure that can lead to interpretable statistical learning models — *shape constraints* in nonparametric statistics and *sparsity* in high-dimensional statistics:

Shape constraints Shape constraints are usually imposed in nonparametric statistical inference and estimation to restrict the feasible region of functions. Some examples of shape constraints can be monotonicity, convexity, log-concavity, unimodality, etc [103]. Shape constraints are usually motivated by either application-specific assumptions or existing theory, thus providing certain interpretability; they can also help reduce the sample complexity and improve statistical performance [127].

Despite great progress in recent years on the statistical theory of the nonpara-

metric shape-constrained estimators [195, 139], adoption of these methods have been limited by the complexities and the intractability of their underlying convex optimization problems [53]. Although there are some nonconvex approaches available, they cannot certify optimality and thus it is unclear how the optimization error is compared with the statistical error for the estimates.

In the first half of the thesis, we focus on two important problems with shape constraints — subgradient regularized multivariate convex regression (in Chapter 2) and multivariate log-concave density estimation (in Chapter 3). Our goal for these two problems is to develop *scalable* computational methods with *convergence guarantees* via convex approaches.

Sparsity Sparsity is a central concept in high-dimensional statistics and data science, where the model selects a small subset of features from a large pool [115]. It is an important and effective regularization technique that can help provide interpretability, reduce the model complexity and mitigate the overfitting issues especially when the model is overparametrized.

Recently, since the work of [28], there has been a renewed interest in exploring statistical problems with structured sparsity using tools from combinatorial optimization. Sparse linear regression [32, 119] and sparse principal component analysis [20] are among examples where combinatorial optimization methods have been successful. However, the exploration of the combinatorial methods for the Gaussian graphical models and general convex loss functions is still limited.

In the second half of the thesis, we focus on two classes of problems with sparsity — Gaussian graphical models (in Chapter 4) and sparse learning problems with general loss functions (in Chapter 5). Our goal is to develop *scalable* computational methods based on *combinatorial optimization*.

The results of Chapters 2 and 3 are based on papers [52, 53], respectively. Below we provide a brief summary of each chapter.

Chapter 2: Subgradient Regularized Multivariate Convex Regression at Scale

In this chapter, we present new large-scale algorithms for fitting a subgradient regularized multivariate convex regression function to n samples in d dimensions—a key problem in shape constrained nonparametric regression with widespread applications in statistics, engineering and the applied sciences. The infinite-dimensional learning task can be expressed via a convex quadratic program (QP) with $O(nd)$ decision variables and $O(n^2)$ constraints. While instances with n in the lower thousands can be addressed with current algorithms within reasonable runtimes, solving larger problems (e.g., $n \approx 10^4$ or 10^5) is computationally challenging. To this end, we present an active set type algorithm on the dual QP. For computational scalability, we allow for approximate optimization of the reduced sub-problems; and propose randomized augmentation rules for expanding the active set. We derive novel computational guarantees for our algorithms. We demonstrate that our framework can approximately solve instances of the subgradient regularized convex regression problem with $n = 10^5$ and $d = 10$ within minutes; and shows strong computational performance compared to earlier approaches. Our implementation is available in a github repository `ConvexRegression`.

Chapter 3: A New Computational Framework for Log-concave Density Estimation

This chapter considers log-concave density estimation. In Statistics, log-concave density estimation is a central problem within the field of nonparametric inference under shape constraints. Despite great progress in recent years on the statistical theory of the canonical estimator, namely the log-concave maximum likelihood estimator, adoption of this method has been hampered by the complexities of the non-smooth convex optimization problem that underpins its computation. We provide enhanced understanding of the structural properties of this optimization problem,

which motivates the proposal of new algorithms, based on both randomized and Nesterov smoothing, combined with an appropriate integral discretization of increasing accuracy. We prove that these methods enjoy, both with high probability and in expectation, a convergence rate of order $1/T$ up to logarithmic factors on the objective function scale, where T denotes the number of iterations. The benefits of our new computational framework are demonstrated on both synthetic and real data, and our implementation is available in a github repository `LogConcComp` (Log-Concave Computation).

Chapter 4: Guassian Graphical Models: A Scalable Framework Based on Combinatorial Optimization

In this chapter, we consider the well-known Gaussian Graphical Models (GGM) problem, where the goal is to estimate the precision matrix of a p-dimensional Normal distribution, given n samples. We propose a new estimator based on the notion of pseudo-likelihood under certain sparsity assumptions on the precision matrix. However, our estimator uses ℓ_0 regularization to encourage sparsity, unlike most current algorithms which are based on convex optimization. We show our estimator can be written as a Mixed Integer Program (MIP). We provide statistical guarantees for our estimator in terms of estimation and variable selection, and discuss how the use of ℓ_0 penalty improves the behavior. Next, we provide a comprehensive optimization framework including heuristic methods to obtain good feasible solution to the MIP, as well as a specialized nonlinear branch-and-bound method to obtain optimal solutions to our proposed MIP. Our numerical experiments on real and synthetic data show that our estimator is computationally scalable to $p \approx 10{,}000$, and provides faster runtime compared to competing (polynomial-time) methods, while leading to superior statistical performance.

Chapter 5: Safe Screening Procedure within a Specialized Branch-and-Bound Solver for Sparse Learning

In this chapter, we focus on sparse learning problems with a general smooth convex loss function and $\ell_0\ell_2$ regularization. In particular, we present a novel screening procedure to safely fix relaxed variables to 0 or 1 at each node within a specialized Branch-and-Bound (BnB) solver to solve this class of problems. We first establish optimality conditions and dual bounds for the node relaxation subproblems within the BnB framework. We then develop a screening procedure at each node within BnB tree, in combination with the branching procedure. This significantly reduces the optimization cost of subproblems and thus reduces the runtime of the BnB solver (up to 2 times faster). For strong branching rules, we propose an enhanced screening procedure which can substantially reduce the size of search trees and further improve the efficiency.

Funding: The work presented in this theses was funded in part by grants from the Office of Naval Research: ONR-N000141812298, N000142112841, N000142212665, the National Science Foundation: NSF-IIS-1718258, MIT IBM Watson AI Lab and Google Research.

Computing resources: The authors acknowledge MIT SuperCloud and Lincoln Laboratory Supercomputing Center for providing HPC resources that have contributed to the research results reported within this thesis.

Chapter 2

Subgradient Regularized Multivariate Convex Regression at Scale

This chapter is based on [52].

2.1 Introduction

Given n samples $(y_i, \boldsymbol{x}_i)$, $i \in [n] := \{1, \ldots, n\}$ with response $y_i \in \mathbb{R}$ and covariates $\boldsymbol{x}_i \in \mathbb{R}^d$, we consider the task of predicting y using a convex function of $\boldsymbol{x}$. This convex function is unknown and needs to be estimated from the data. This leads rise to the so-called multivariate *convex regression* problem [200, 149] where we minimize the sum of squared residuals

$$\hat{\varphi} = \arg\min_{\varphi \in \mathcal{F}} \ \frac{1}{n} \sum_{i=1}^{n} (y_i - \varphi(\boldsymbol{x}_i))^2 \tag{2.1}$$

over all real-valued convex functions in $\mathbb{R}^d$, denoted by $\mathcal{F}$. Above, $\varphi(\boldsymbol{x}_i)$ is the value of the convex function φ at point $\boldsymbol{x}_i$.

In the special case where $\varphi(\boldsymbol{x}) = \boldsymbol{x}^\top \boldsymbol{\beta}$ is a linear function, with unknown regression coefficients $\boldsymbol{\beta}$, criterion (2.1) leads to the well-known least squares problem. Problem (2.1) is an instance of shape constrained nonparametric regression [192] — here we learn the underlying function φ under a qualitative shape constraint such

as convexity. The topic of function estimation under shape constraints has received significant attention in recent years — see for example, the special issue in *Statistical Science* [196] for a nice overview.

Convex regression is widely used in economics, operations research, statistical learning and engineering applications. In economics applications, for example, convexity/concavity arise in modeling utility and production functions, consumer preferences [210, 127], among others. In some stochastic optimization problems, value functions are taken to be convex [202]. See also the works of [112, 219, 14] for other important applications of convex regression.

There is a rich body of work in statistics studying different (statistical) methodological aspects of convex regression [200, 149, 14, 110, 111, 139, 140, 141]. However, the challenges associated with computing the convex regression estimator limit our empirical understanding of this estimator and its usage in practice. More recently, there is a growing interest in developing efficient algorithms for this optimization problem — see for e.g. [158, 11, 31, 150]. The focus of this chapter is to further advance the computational frontiers of convex regression.

We note that the infinite-dimensional Problem (2.1) can be reduced [200, 158] to a finite dimensional convex quadratic program (QP):

$$\begin{aligned} \underset{\phi_1,\ldots,\phi_n;\boldsymbol{\xi}_1,\ldots,\boldsymbol{\xi}_n}{\text{minimize}} \quad & \frac{1}{n}\sum_{i=1}^{n}(y_i-\phi_i)^2 \\ \text{s.t.} \quad & \phi_j-\phi_i \geq \langle \boldsymbol{x}_j-\boldsymbol{x}_i, \boldsymbol{\xi}_i\rangle, \quad \forall (i,j)\in\Omega \end{aligned} \tag{2.2}$$

where $\phi_1,\ldots,\phi_n \in \mathbb{R}$, $\boldsymbol{\xi}_1,\ldots,\boldsymbol{\xi}_n \in \mathbb{R}^d$, and $\Omega := \{(i,j) : 1 \leq i,j \leq n, i \neq j\}$. In (2.2), $\phi_i = \varphi(\boldsymbol{x}_i)$ and $\boldsymbol{\xi}_i$ is a subgradient of $\varphi(\boldsymbol{x})$ at $\boldsymbol{x} = \boldsymbol{x}_i$. Problem (2.2) has $O(nd)$ variables and $O(n^2)$ constraints and becomes computationally challenging when n is large. For the convex regression problem to be statistically meaningful [139, 158], we consider cases with $n \gg d$ (and number of features $d \sim \log n$ to be small). Off-the-shelf interior point methods [200] for (2.2) are limited to instances where n is at most a few hundred. [11] consider a regularized version of (2.2) (i.e., (P_0), below) and propose parallel algorithms to solve instances with $n \approx 1{,}600$ leveraging

commercial QP solvers (like Mosek). [158] use an alternating direction method of multipliers (ADMM) [39]-based algorithm that can address problems up to $n \approx 3{,}000$. Recently, [150] propose a different ADMM method and also a proximal augmented Lagrangian method where the subproblems are solved by the semismooth Newton method—they address instances with $n \approx 3{,}000$. Algorithms based on nonconvex optimization have been proposed to learn convex functions that are representable as a piecewise maximum of k-many hyperplanes [112, 14, 96]—these are interesting approaches, but they may not lead to an optimal solution for the convex regression convex optimization problem. Recently, [31] present a cutting plane or constraint generation-type algorithm for (2.2): At every iteration, they solve a reduced QP by considering a subset of constraints. Leveraging capabilities of commercial solvers (e.g., Gurobi), this can approximately solve instances of (2.2) with $n \approx 10^4 - 10^5$. In this chapter, we also present an active-set type method, but our approach differs from [31], as we discuss below. We also establish novel computational guarantees for our proposed approach.

The convex least squares estimator (2.2) may lead to undesirable statistical properties when $\boldsymbol{x}$ is close to the boundary of the convex hull of $\{\boldsymbol{x}_i\}_1^n$ [158, 13]. This problem can be improved by considering a subgradient regularized version of (2.2) given by:

$$\begin{array}{ll} \underset{\phi_1,\ldots,\phi_n;\boldsymbol{\xi}_1,\ldots,\boldsymbol{\xi}_n}{\text{minimize}} & \dfrac{1}{n}\displaystyle\sum_{i=1}^{n}(y_i-\phi_i)^2+\dfrac{\rho}{n}\sum_{i=1}^{n}\|\boldsymbol{\xi}_i\|^2 \\ \text{s.t.} & \phi_j-\phi_i \geq \langle \boldsymbol{x}_j-\boldsymbol{x}_i, \boldsymbol{\xi}_i\rangle, \quad \forall (i,j)\in\Omega, \end{array} \tag{P_0}$$

where we impose an additional ℓ_2-based regularization on the subgradients $\{\boldsymbol{\xi}_i\}_1^n$ of φ; with $\rho > 0$ being the regularization parameter. Formulation (P_0) also appears in [11]. Statistical estimation error properties of a form of subgradient regularized convex least squares estimator appear in [158]. In the special case, where $\varphi(\boldsymbol{x}) = \boldsymbol{x}^\top\boldsymbol{\beta}$ is a linear function, (P_0) leads to the popular ridge regression estimator (i.e., least squares with an additional squared ℓ_2 penalty on $\boldsymbol{\beta}$). In ridge regression a nonzero penalty on $\|\boldsymbol{\beta}\|_2^2$ often leads to improved statistical performance over vanilla least squares. Similarly,

in convex regression, a value of $\rho > 0$ can result in better statistical estimation error compared to the unregularized estimator with $\rho = 0$—See Section 2.5 for numerical support on a collection of datasets.

Our approach: In this chapter, we focus on solving (P_0) for $\rho > 0$. Problem (P_0) has a strongly convex objective function in the decision variables $(\{\phi_i\}_1^n, \{\boldsymbol{\xi}_i\}_1^n)$ — its Lagrangian dual (see (D) below) is a convex QP with $O(n^2)$ variables over the nonpositive orthant. We present large scale algorithms for this dual and study their computational guarantees. The large number of variables poses computational challenges for full-gradient-based optimization methods as soon as n becomes larger than a few thousand. However, we anticipate that $O(n)$-many of the constraints in (P_0) will suffice. We draw inspiration from the one dimensional case ($d = 1$), an observation that was also used by [31]. Hence, we use methods inspired by constraint generation [34], which we also refer to (with an abuse of terminology) as active set type methods [25]. Every step of our algorithm considers a reduced dual problem where the decision variables, informally speaking, correspond to a subset of the primal constraints. The vanilla version of this active set method, which solves the reduced dual problem to *optimality*[1], becomes expensive when n and/or the size of the active set becomes large especially if one were to perform several active-set iterations. We propose improved algorithms that perform inexact (or approximate) optimization for this reduced problem initially and then increase the optimization accuracy at a later stage. Upon solving the reduced problem (exactly or inexactly), we examine optimality conditions for the full problem; and include additional variables into the dual problem, if necessary.

To augment the current active-set, greedy deterministic augmentation rules that scan all $O(n^2)$-constraints, become computationally expensive — therefore, we use randomized rules, which leads to important cost savings. These randomized augmentation rules extend the random-then-greedy selection strategies proposed by [154] in the context of Gradient Boosting Machines [92]. Our approach operates on the dual

[1]This is similar to the method of [31] who consider a constraint generation method for (2.2) where the reduced sub-problems are solved to optimality.

and results in a dual feasible — we show how this leads to a primal feasible solution, delivering a duality gap certificate.

We establish a novel linear convergence rate of our algorithm (in terms of outer iterations) on the dual, which is not strongly convex. Our guarantees apply to both exact/inexact optimization of the reduced problem; and both deterministic and randomized augmentation rules. As we focus on large scale problems (e.g. $n \geq 10,000$), inexact optimization of the reduced sub-problem and randomized augmentation rules play a key role in computational efficiency. As we carefully exploit problem-structure, our standalone algorithms enjoy a low memory footprint and can approximately solve instances of subgradient regularized convex regression with $n \approx 10^5$ and $d = 10$ in minutes. Numerical comparisons suggest that on several datasets, our approach appears to notably outperform earlier algorithms in solving (P_0) for values of $\rho > 0$ that result in good statistical performance. Since our approach is based on the smooth dual of (P_0), the performance of our algorithm would deteriorate when ρ is numerically very close to zero. In particular, our approach may not be suitable for obtaining a high-accuracy solution to the unregularized problem (2.2).

Organization of chapter: Section 2.2 presents both primal and dual formulations of the full problem (P_0); and a first order method on the dual. Section 2.3 presents active-set type algorithms, augmentation rules and associated computational guarantees. Section 2.4 discusses computing duality gap certificates. Section 2.5 presents numerical experiments. Some technical details are relegated to Section 2.A to improve readability.

Notations: For convenience, we list some notations used throughout the chapter. We denote the set $\{1, 2, \ldots, n\}$ by $[n]$. The cardinality of a set W is denoted by $|W|$. We denote by $\mathbb{R}_+$, $\mathbb{R}_{++}$ the set of nonnegative and positive real numbers (respectively). A similar notation applies for $\mathbb{R}_-$ and $\mathbb{R}_{--}$. Symbols $\mathbf{1}_n$, $\boldsymbol{e}_i$ and $\boldsymbol{I}$ denote: a vector of length n of all ones, the i-th standard basis element and the identity matrix (respectively). span(A) denotes the linear space generated by the vectors in the set A. For a matrix $\boldsymbol{B}$, let vec$(\boldsymbol{B})$ denote a vectorized version of $\boldsymbol{B}$. The largest singular

value of a matrix $\boldsymbol{B}$ is denoted by $\lambda_{\max}(\boldsymbol{B})$. We use $\|\cdot\|$ to denote the Euclidean norm of a vector and the spectral norm of a matrix. Finally, $\partial f(\boldsymbol{x})$ denotes the subdifferential (set of subgradients) of f at $\boldsymbol{x}$.

2.2 Primal and Dual Formulations

We introduce some notation to rewrite Problem (P_0) compactly. Let $\boldsymbol{y} = [y_1, \ldots, y_n]^\top \in \mathbb{R}^n$, $\boldsymbol{X} = [\boldsymbol{x}_1^\top, \ldots, \boldsymbol{x}_n^\top] \in \mathbb{R}^{n \times d}$, $\boldsymbol{\phi} = [\phi_1, \ldots, \phi_n]^\top \in \mathbb{R}^n$, and $\boldsymbol{\xi} = \mathrm{vec}([\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_n]) \in \mathbb{R}^{nd}$. We define $\boldsymbol{A} \in \mathbb{R}^{n(n-1) \times n}$ and $\boldsymbol{B} \in \mathbb{R}^{n(n-1) \times nd}$ such that the rows of $\boldsymbol{A}\boldsymbol{\phi} + \boldsymbol{B}\boldsymbol{\xi}$ correspond to the constraints $\phi_j - \phi_i - \langle \boldsymbol{x}_j - \boldsymbol{x}_i, \boldsymbol{\xi}_i \rangle$ for $(i, j) \in \Omega$. Hence (P_0) is equivalent to:

$$\underset{\boldsymbol{\phi}, \boldsymbol{\xi}}{\text{minimize}} \quad f(\boldsymbol{\phi}, \boldsymbol{\xi}) := \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{\phi}\|^2 + \frac{\rho}{2}\|\boldsymbol{\xi}\|^2 \qquad \text{s.t.} \quad \boldsymbol{A}\boldsymbol{\phi} + \boldsymbol{B}\boldsymbol{\xi} \geq \boldsymbol{0}, \tag{P}$$

where $\boldsymbol{A}\boldsymbol{\phi} + \boldsymbol{B}\boldsymbol{\xi} \geq \boldsymbol{0}$ denotes componentwise inequality. Due to strong convexity of the objective, (P) has a unique minimizer denoted by $(\boldsymbol{\phi}^\star, \boldsymbol{\xi}^\star)$. While (P) has a large number (i.e., $O(n^2)$) of constraints, given that the affine hull of the data points has full dimension[2], i.e. $\mathrm{span}(\{\boldsymbol{x}_i - \boldsymbol{x}_j\}_{j \neq i}) = \mathbb{R}^d$, it can be shown that the constraint matrix $\boldsymbol{C} = \begin{bmatrix} \boldsymbol{A} & \boldsymbol{B} \end{bmatrix} \in \mathbb{R}^{n(n-1) \times (n+nd)}$ is of rank $O(nd)$. This serves as a motivation for our active-set approach.

The Lagrangian dual of (P) is equivalent to the following convex problem

$$L^\star = \underset{\boldsymbol{\lambda} \in \mathbb{R}^{n(n-1)}}{\text{minimize}} \quad L(\boldsymbol{\lambda}) := \frac{1}{2\rho}\boldsymbol{\lambda}^\top(\rho\boldsymbol{A}\boldsymbol{A}^\top + \boldsymbol{B}\boldsymbol{B}^\top)\boldsymbol{\lambda} - \boldsymbol{y}^\top\boldsymbol{A}^\top\boldsymbol{\lambda} \quad \text{s.t.} \quad \boldsymbol{\lambda} \leq 0. \tag{D}$$

Definition 2.1. *A convex function f is σ-smooth if it is continuously differentiable with σ-Lipschitz gradient; f is μ-strongly convex if $f(\boldsymbol{x}) - \frac{\mu}{2}\boldsymbol{x}^\top\boldsymbol{x}$ is convex.*

Note that $\boldsymbol{\lambda} \mapsto L(\boldsymbol{\lambda})$ is not strongly convex, but it is σ-smooth, where σ is the maximum eigenvalue of the matrix $\boldsymbol{Q} := \boldsymbol{A}\boldsymbol{A}^\top + \frac{1}{\rho}\boldsymbol{B}\boldsymbol{B}^\top$.

[2]This occurs with probability one if the covariates are drawn from a continuous distribution.

Unlike the primal (P), the dual problem (D) is amenable to proximal gradient methods [175, 19] (PGD). Other gradient based methods like accelerated proximal gradient methods (APG) [19, 175], the limited-memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS) method [152] (for example), may also be used; and they work well in our numerical experience. However, every iteration of PGD requires computing the gradient $\nabla L(\boldsymbol{\lambda}) \in \mathbb{R}^{n(n-1)}$. While an unstructured gradient computation will cost $O(n^4)$, exploiting the structure of $\boldsymbol{A}, \boldsymbol{B}$, this cost can be reduced to $O(n^2 d)$, allowing us to scale these algorithms for instances with $n \approx 3,000$. Such matrix-vector multiplications can be used to estimate σ via the power method or backtracking line-search [19]. When L is σ-smooth, PGD enjoys a standard sublinear convergence rate $O(1/t)$ [19]. With an additional strong convexity assumption PGD is known to converge at a linear rate [173]. Note $L(\boldsymbol{\lambda})$ is not strongly convex. However, $L(\boldsymbol{\lambda})$ satisfies the Polyak-Łojasiewicz condition [129, 170] under which PGD converges at a linear rate. We note more general convergence results under error bound conditions can be found in [155, 156, 147, 37].

2.3 Active Set Type Algorithms

As (D) has $O(n^2)$ variables, the proximal gradient method (owing to full gradient computations) becomes prohibitively expensive when n becomes larger than a few thousand. However, as discussed earlier, we expect only $O(n)$-many of these variables to be nonzero at an optimal solution—motivating the use of a constraint-generation/active set-type method on the primal (P), which relates to a column-generation type method on the dual (D).

Constraint generation is traditionally used in the context of solving large-scale linear programs [64, 34]. When used in the context of the QP (P), as done in [31], we start with a reduced problem with a small subset of constraints in (P). With a slight abuse of nomenclature, we refer to these constraints as the active set[3]. After obtaining an optimal dual solution to this reduced problem, the traditional form of constraint

[3]Our usage of "active set" differs from the active set method for solving a QP, as discussed in [176].

generation will augment the active set with some dual variables that correspond to the violated primal constraints (if any) and re-solve the problem on the expanded set of constraints. We mention two shortcomings of this approach: (a) Solving the reduced problem to optimality becomes expensive (especially when the active set becomes large and/or if several iterations of constraint-generation is needed); and (b) finding variables to be appended to the active set has a large cost of $O(n^2 d)$ operations.

To circumvent these shortcomings, we propose modifications to the above constraint generation or active set method. To address (a), we solve the reduced subproblem inexactly (e.g., by taking a few iterations of the proximal gradient method). To address (b), we consider randomized rules to reduce the cost of augmenting the active set from $O(n^2 d)$ to $O(nd)$ (for example). We show that our proposed algorithm converges; and does so with a linear convergence rate in the outer iterations.

2.3.1 Properties of the reduced problem

Let $W \subseteq \Omega$ index a subset of the constraints in (P); and consider the reduced primal:

$$\underset{\boldsymbol{\phi},\boldsymbol{\xi}}{\text{minimize}} \quad f(\boldsymbol{\phi},\boldsymbol{\xi}) = \frac{1}{2}\|\boldsymbol{y}-\boldsymbol{\phi}\|^2 + \frac{\rho}{2}\|\boldsymbol{\xi}\|^2 \qquad \text{s.t.} \quad \boldsymbol{A}_W\boldsymbol{\phi} + \boldsymbol{B}_W\boldsymbol{\xi} \geq \boldsymbol{0} \qquad (P_W)$$

where, $\boldsymbol{A}_W$ (and $\boldsymbol{B}_W$) denotes matrix $\boldsymbol{A}$ (and $\boldsymbol{B}$) restricted to rows indexed by W.

In the rest of the chapter, we will use W as a subscript for vectors or matrices whose size changes with W, and use W (or $[W]$) as superscript for vectors or matrices whose size does not change with W. When $W = \Omega$, the relaxed problem is the original problem, and we drop the use of Ω as subscript and/or superscript for notational convenience.

We consider solving the dual of (P_W). Proposition 2.1 presents some of its properties.

Proposition 2.1. *The Lagrangian dual of* (P_W) *is given by:*

$$\min_{\boldsymbol{\lambda}_W} \quad L_W(\boldsymbol{\lambda}_W) := \tfrac{1}{2\rho}\boldsymbol{\lambda}_W^\top(\rho\boldsymbol{A}_W\boldsymbol{A}_W^\top + \boldsymbol{B}_W\boldsymbol{B}_W^\top)\boldsymbol{\lambda}_W - \boldsymbol{y}^\top\boldsymbol{A}_W^\top\boldsymbol{\lambda}_W \quad \text{s.t.} \quad \boldsymbol{\lambda}_W \leq \boldsymbol{0}. \quad (D_W)$$

Let $L_W^\star$ be the optimal objective value for (D_W). The objective function $L_W(\cdot) : \mathbb{R}^{|W|} \to \mathbb{R}$ is σ_W-smooth for some $\sigma_W \leq \sigma$. The set of all optimal solutions to (D_W) is a polyhedron of the form

$$\Lambda_W^\star = \left\{ \boldsymbol{\lambda}_W \in \mathbb{R}^{|W|} : \boldsymbol{A}_W^\top \boldsymbol{\lambda}_W = \boldsymbol{s}_A^W, \frac{1}{\sqrt{\rho}} \boldsymbol{B}_W^\top \boldsymbol{\lambda}_W = \boldsymbol{s}_B^W, \boldsymbol{\lambda}_W \leq 0 \right\} \tag{2.3}$$

where, $\boldsymbol{s}_A^W = \boldsymbol{y} - \boldsymbol{\phi}_\star^W, \boldsymbol{s}_B = -\sqrt{\rho}\boldsymbol{\xi}_\star^W$; and $(\boldsymbol{\phi}_\star^W, \boldsymbol{\xi}_\star^W)$ is the unique optimal solution to the reduced primal problem (P_W).

Proof Sketch. The Lagrangian dual is obtained by computing $\min_{\boldsymbol{\xi},\boldsymbol{\phi}} \mathcal{L}(\boldsymbol{\phi}, \boldsymbol{\xi}; \boldsymbol{\lambda}_W)$, where $\mathcal{L}(\boldsymbol{\phi}, \boldsymbol{\xi}; \boldsymbol{\lambda}_W) = f(\boldsymbol{\phi}, \boldsymbol{\xi}) + \boldsymbol{\lambda}_W^\top (\boldsymbol{A}_W \boldsymbol{\phi} + \boldsymbol{B}_W \boldsymbol{\xi})$ for any $\boldsymbol{\lambda}_W \leq 0$. The optimality (KKT) conditions for $\boldsymbol{\phi}, \boldsymbol{\xi}$ are $\boldsymbol{\phi}_\star^W = \boldsymbol{y} + \boldsymbol{A}_W^\top \boldsymbol{\lambda}_W$ and $\boldsymbol{\xi}_\star^W = \rho \boldsymbol{B}_W^\top \boldsymbol{\lambda}_W$. Plugging these equations into $\mathcal{L}$ and flipping the sign of the function lead to the objective L_W. As $\boldsymbol{\lambda}_W \mapsto L_W(\boldsymbol{\lambda}_W)$ is a quadratic function, it is σ_W-smooth with σ_W being the maximum eigenvalue of the Hessian matrix. Since the Hessian of the reduced dual problem is a submatrix of that of the original dual problem (corresponding to W out of Ω), it follows that $\sigma_W \leq \sigma$. The formulation of the polyhedral set follows from the KKT conditions [40]. □

PGD is readily applicable to (D_W). The per iteration cost is $O(|W|d)$, which is much smaller than $O(n^2 d)$ (as $|W| \sim n$). Solving the reduced problem (D_W) is usually much faster than solving the full problem (D) when $|W| \ll |\Omega|$.

2.3.2 Augmentation Rules

For any $W \subseteq \Omega$, given a feasible solution[4] $\boldsymbol{\lambda}_W \in \mathbb{R}_-^{|W|}$ to the reduced dual (D_W), we can construct its corresponding primal variables $(\boldsymbol{\phi}^W, \boldsymbol{\xi}^W)$ for (P_W), by making use of the KKT conditions:

$$(\boldsymbol{\phi}^W, \boldsymbol{\xi}^W) = (\boldsymbol{y} - \boldsymbol{A}_W^\top \boldsymbol{\lambda}_W, -\frac{1}{\rho} \boldsymbol{B}_W^\top \boldsymbol{\lambda}_W) \in \mathbb{R}^n \times \mathbb{R}^{nd}. \tag{2.4}$$

[4]This can be obtained by solving (D_W) exactly (i.e., to optimality) or inexactly (i.e., approximately).

Notice that for a general dual variable $\boldsymbol{\lambda}_W \in \mathbb{R}_-^{|W|}$, the primal variables obtained via (2.4) may not be feasible for the reduced primal problem (P_W). Below we discuss some rules for augmenting the current set of constraints (i.e., the active set).

After obtaining $(\boldsymbol{\phi}^W, \boldsymbol{\xi}^W)$, we check if it is feasible for (P)—that is, we verify if each component of $\boldsymbol{A}\boldsymbol{\phi}^W + \boldsymbol{B}\boldsymbol{\xi}^W$ (denoted by $v_{(i,j)}$) is nonnegative:

$$v_{(i,j)} = \phi_j^W - \phi_i^W - \langle \boldsymbol{x}_j - \boldsymbol{x}_i, \boldsymbol{\xi}_i^W \rangle \geq 0 \quad \forall (i,j) \in \Omega. \tag{2.5}$$

To this end, it is helpful to decompose the $O(n^2)$ constraints into n blocks

$$\Omega = \bigcup_i \Omega_{i\cdot} = \bigcup_j \Omega_{\cdot j},$$

where $\Omega_{i\cdot} = \{(i,j) : j \neq i, 1 \leq j \leq n\}$ and $\Omega_{\cdot j} = \{(i,j) : i \neq j, 1 \leq i \leq n\}$. Similarly, we define the slices $W_{i\cdot}$ and $W_{\cdot i}$ for all i. The two decompositions $\{\Omega_{i\cdot}\}_1^n$ and $\{\Omega_{\cdot j}\}_1^n$ have different geometric interpretations. Now define points $P_j = \{\boldsymbol{x}_j, \phi_j^W\}$ and hyperplanes $H_i : y = \langle \boldsymbol{x} - \boldsymbol{x}_i, \boldsymbol{\xi}_i^W \rangle + \phi_i^W$ in $\mathbb{R}^{d+1}$. Note that each point P_i lies on the hyperplane H_i, and $v_{(i,j)}$ denotes the vertical distance between P_j and H_i. For each i, the nonnegativity of $v_{(i,j)}$ for all $(i,j) \in \Omega_{i\cdot}$, checks if the hyperplane H_i lies below each point P_j, i.e. if H_i is a supporting hyperplane of the points $\{P_j\}_1^n$. On the other hand, for each j, the nonnegativity of $v_{(i,j)}$ for all $(i,j) \in \Omega_{\cdot j}$, checks if P_j lies above each hyperplane H_i—i.e., if P_j lies above the piecewise maximum of these hyperplanes. The quantity $|\min_i v_{(i,j)}|$ is the amount by which P_j lies below the piecewise maximum.

In Section 2.4 we see that the block decomposition interpretation is useful in obtaining a primal feasible solution. Next we discuss a deterministic augmentation rule — due to its high computational cost, we subsequently present randomized augmentation rules — both of which make use of the above decomposition of Ω.

2.3.2.1 Deterministic Augmentation Rule

We first present a simple greedy-like deterministic augmentation rule:

Rule 1. **Greedy within each Block:** For each block $\Omega_{i\cdot}$ (or $\Omega_{\cdot j}$), choose P pairs with the smallest $v_{(i,j)}$-values among $\Omega_{i\cdot}\backslash W_{i\cdot}$ (or $\Omega_{\cdot j}\backslash W_{\cdot j}$). From these P indices, we add to the current active set W, only those with negative $v_{(i,j)}$-values.

Rule 1 evaluates $O(n^2)$ constraints with computational cost $O(n^2d + d|W|)$ and augments W by at most nP such constraints. For every block, obtaining the largest P violations can be done via a partial sort—leading to a total cost of $O(n^2d + nP\log P)$ for n blocks[5]. As a result, with this greedy rule, the augmentation becomes a major computational bottleneck for the active-set type method. This motivates the randomized augmentation rules discussed below.

2.3.2.2 Randomized Augmentation Rules

We present four randomized rules that sample a small subset of the indices in $\Omega \setminus W$ instead of performing a computationally intensive full scan across $O(n^2)$ indices as in Rule 1.

Rule 2. Random: Uniformly sample K indices in $\Omega\backslash W$—the cost of computing the corresponding $v_{(i,j)}$-values is $O(Kd + d|W|)$.

Rule 3. Random within each Block: For each block $\Omega_{i\cdot}$ (or $\Omega_{\cdot j}$), uniformly sample P elements in $\Omega_{i\cdot}\backslash W_{i\cdot}$ (or $\Omega_{\cdot j}\backslash W_{\cdot j}$); the total computational cost is $O(nPd + d|W|)$.

Rule 4. Random then Greedy: Uniformly sample M-many (i,j)-pairs from $\Omega\backslash W$, and from these pairs choose the K pairs with the smallest $v_{(i,j)}$-values. Computing the M values of $v_{(i,j)}$ cost $O(Md + d|W|)$, and the greedy selection step costs $M + K\log K$. The total computational cost is $O(Md + d|W|)$.

Rule 5. Random Blocks then Greedy within each Block: Uniformly sample G groups from $\{\Omega_{i\cdot}\}_{i=1}^n$ (or $\{\Omega_{\cdot j}\}_{j=1}^n$) and for each group, choose the P pairs that have the smallest $v_{(i,j)}$ values with $(i,j) \in \Omega_{i\cdot}\backslash W_{i\cdot}$ (or $\Omega_{\cdot j}\backslash W_{\cdot j}$). Similar to Rule 4, the total computational cost is $O(Gnd + d|W|)$.

Note that the above rules lead to a set of indices denoted by Δ'. From these

[5]A similar augmentation rule with $P = 1$ is used in [31]

candidates, we only select those with negative $v_{(i,j)}$-values, which are then appended to the current active set W. That is, if $\Delta = \{(i,j) \in \Delta' : v_{(i,j)} < 0\}$ then we set $W \leftarrow W \cup \Delta$.

Of the above, Rules 4 and 5 are inspired by, but different from the random-then-greedy coordinate descent procedure [154], proposed in the context of Gradient Boosting.

2.3.2.3 Norms Associated with the Augmentation Rules

The computational guarantees of our active set-type algorithms depend upon certain norms induced by the aforementioned augmentation rules. We first present some notation that we will use in our analysis.

Definition 2.2. *Given a vector $\boldsymbol{\theta}$, an index set S, and $k \leq |S|$, let $\mathcal{G}[S, k, \boldsymbol{\theta}]$ be the set of k elements with the largest values of $|\theta_\omega|$ for $\omega \in S$. Given a pair (S, k) as above, we let $\mathcal{U}[S, k]$ denote the set of k uniformly subsampled indices from the set S.*

Definition 2.3. *Given a vector $\boldsymbol{\theta} \in \mathbb{R}^{n(n-1)}$ indexed by Ω, let $\{S_i\}_1^n$ be disjoint subsets of Ω, with $\bar{S} = \cup_{i\in[n]} S_i$. Given positive integers P, K, M, G, we define $\delta_1(\boldsymbol{\theta}, \{S_i\})$, $\ldots, \delta_5(\boldsymbol{\theta}, \{S_i\}) \subset \Omega$ as follows:*

$$\delta_1(\boldsymbol{\theta}, \{S_i\}) = \bigcup_{i=1}^{n} \mathcal{G}[S_i, P, \boldsymbol{\theta}], \quad \delta_2(\boldsymbol{\theta}, \{S_i\}) = \mathcal{U}[\bar{S}, K], \quad \delta_3(\boldsymbol{\theta}, \{S_i\}) = \bigcup_{i=1}^{n} \mathcal{U}[S_i, P]$$
$$\delta_4(\boldsymbol{\theta}, \{S_i\}) = \mathcal{G}[\mathcal{U}[\bar{S}, M], K, \boldsymbol{\theta}], \qquad \delta_5(\boldsymbol{\theta}, \{S_i\}) = \bigcup_{i \in \mathcal{U}[[n], G]} \mathcal{G}[S_i, P, \boldsymbol{\theta}].$$

With the above notation, we can express the indices to be augmented as a function of the violations $\boldsymbol{v}$, a vectorized representation of the entries $\{v_{(i,j)}\}_{(i,j)\in\Omega}$ (2.5). Let $\Delta'_{\{\ell\}}$ denote the pairs selected by Rule ℓ, and $\Delta_{\{\ell\}} = \{\omega \in \Delta'_{\{\ell\}} : v_{(i,j)} < 0\} \subseteq \Delta'_{\{\ell\}}$ be the final set to be added to W. Let $\tilde{\boldsymbol{v}}$ be a vector having the same length as $\boldsymbol{v}$, with its entries given by $\tilde{v}_{(i,j)} = \min\{v_{(i,j)}, 0\}$. Then it is easy to verify that $\Delta'_{\{\ell\}}$ can be written as $\delta_\ell(\tilde{\boldsymbol{v}}, \{S_i\})$ with $S_i = \Omega_{i\cdot} \setminus W_{i\cdot}$ or $S_i = \Omega_{\cdot i} \setminus W_{\cdot i}$.

Definition 2.4. *Given $\boldsymbol{\theta} \in \mathbb{R}^{n(n-1)}$, let $\{\Omega_i\}_1^n$ be either $\{\Omega_{i\cdot}\}_1^n$ or $\{\Omega_{\cdot i}\}_1^n$. Define*

$$\|\boldsymbol{\theta}\|_{\{\ell\}} = \left(\mathbb{E}\left[\sum_{\omega \in \delta_\ell(\boldsymbol{\theta}, \{\Omega_i\})} |\theta_\omega|^2 \right] \right)^{1/2} \quad \text{for} \quad \ell \in \{1, \ldots, 5\}. \tag{2.6}$$

For $\ell = 1$, there is no randomness in δ_ℓ, so the expectation can be removed. For $\ell \geq 2$, the expectation is taken over the randomness arising from the selection operator $\mathcal{U}$ (cf Definition 2.2).

Lemma 2.1 shows that the expression in display (2.6) leads to a norm on $\mathbb{R}^{n(n-1)}$. Furthermore, the norm equivalence constants appearing in (2.7) determine the convergence rates of Algorithm 2.1 (see Theorem 2.1).

Lemma 2.1. *$\|\boldsymbol{\theta}\|_{\{\ell\}}$ defined in (2.6) is a norm on $\mathbb{R}^{n(n-1)}$. Furthermore, the constants $\alpha_{\{\ell\}}, \beta_{\{\ell\}}$ listed in Table 2.1 satisfy that for all $\boldsymbol{x} \in \mathbb{R}^{n(n-1)}$*

$$\alpha_{\{\ell\}} \|\boldsymbol{x}\|_{\{\ell\}}^2 \geq \|\boldsymbol{x}\|_2^2 \geq \beta_{\{\ell\}} \|\boldsymbol{x}\|_{\{\ell\}}^2. \tag{2.7}$$

The proof of Lemma 2.1 appears in Section 2.A.1.

Table 2.1 presents a summary of some key characteristics of the five rules.

Table 2.1: Summary of some properties of the augmentation rules. Recall that $\Delta'_{\{\ell\}}$ denotes the pairs selected as per Rule ℓ. The number of candidates to be augmented to the current active set depends upon the signs of $v_{(i,j)}$s; and is of size at most $|\Delta'_{\{\ell\}}|$. Here, *Augmentation Cost* is the cost of obtaining $\Delta'_{\{\ell\}}$. We present estimates of the norm-equivalence constants $\alpha_{\{\ell\}}, \beta_{\{\ell\}}$ (2.7). For notational convenience, we assume that $W = \varnothing$—for a nonempty W, we can replace Ω_i with $\Omega_i \backslash W_i$.

Rule	$\Delta'_{\{\ell\}}$	$\lvert\Delta'_{\{\ell\}}\rvert$	Augmentation Cost	$\alpha_{\{\ell\}}$	$\beta_{\{\ell\}}$
1	$\cup_{i=1}^n \mathcal{G}[\Omega_i, P, \tilde{\boldsymbol{v}}]$	nP	$O(n^2 d + d\lvert W\rvert)$	$\frac{n-1}{P}$	1
2	$\mathcal{U}[\Omega, K]$	K	$O(Kd + d\lvert W\rvert)$	$\frac{n(n-1)}{K}$	$\frac{n(n-1)}{K}$
3	$\cup_{i=1}^n \mathcal{U}[\Omega_i, P]$	nP	$O(nPd + d\lvert W\rvert)$	$\frac{n-1}{P}$	$\frac{n-1}{P}$
4	$\mathcal{G}[\mathcal{U}[\Omega, M], K, \tilde{\boldsymbol{v}}]$	K	$O(Md + d\lvert W\rvert)$	$\frac{n(n-1)}{K}$	$\frac{n(n-1)}{M}$
5	$\cup_{i \in \mathcal{U}[[n],G]} \mathcal{G}[\Omega_i, P, \tilde{\boldsymbol{v}}]$	GP	$O(Gnd + d\lvert W\rvert)$	$\frac{n(n-1)}{GP}$	$\frac{n}{G}$

2.3.3 Active set method with inexact optimization of sub-problems

Once we augment the active set (based on Rules 1–5), we solve the updated reduced QP either exactly or inexactly. We then identify additional constraints to be added to the current active set; and continue till some convergence criteria is satisfied. Our algorithm is summarised below:

Algorithm 2.1 An Active Set Type Method on the Dual (D)

Input: Initialize with W^0 and $\boldsymbol{\lambda}^0$.

1: **for** $m = 0, 1, \ldots$ **do**

2: Step 1: Obtain a feasible solution $\boldsymbol{\lambda}_{W^m}$ for (D_{W^m}) by either (a) solving (D_{W^m}) to optimality (i.e., exactly) or (b) solving (D_{W^m}) inexactly via a few steps of proximal gradient descent[6].

3: Step 2: Compute $\boldsymbol{\phi}^{W^m} = \boldsymbol{y} - \boldsymbol{A}_{W^m}^\top \boldsymbol{\lambda}_{W^m}$, $\boldsymbol{\xi}^{W^m} = -\frac{1}{\rho}\boldsymbol{B}_{W^m}^\top \boldsymbol{\lambda}_{W^m}$ as per (2.4).

4: Step 3: Use one of the Rules 1–5 to augment W^m to obtain W^{m+1}.

5: **end for**

In what follows, we call Algorithm 2.1 with option (a) [Step 1] the *Exact Active Set* method (EAS); and option (b) [Step 1] the *Active Set Gradient Descent* (ASGD) method.

If the size of the active set remains sufficiently small, interior point solvers (including heavy-duty commercial solvers like Gurobi, Mosek) may be used for full minimization of the reduced problem, as long as the number of active set iterations remain small. However, for larger problems (e.g., when $n \approx 10^4 - 10^5$), first order methods may be preferable due to warm-start capabilities (across active sets) and low memory requirements (by exploiting problem-structure). Furthermore, they also allow for inexact computation for the sub-problems—see numerical experiments in Section 2.5 showing the important benefits in approximate optimization.

Theorem 2.1 establishes that Algorithm 2.1 requires at most $m = O(\log(1/\epsilon))$-many outer iterations to deliver an ϵ-optimal dual solution for Problem (D):

[6]Other choices are also possible, as discussed in Remark 2.2.

Theorem 2.1. *For a given $\boldsymbol{\lambda}^0$ and W^0, let $\boldsymbol{\lambda}^m = \boldsymbol{\lambda}^{[W_m]}$ (for $m \geq 1$) be the sequence generated by Algorithm 2.1. Then, for either EAS or ASGD with augmentation Rule ℓ, we have:*

$$\mathbb{E}[L(\boldsymbol{\lambda}^m) - L^\star] \leq \left(1 - \frac{\mu}{\sigma\alpha_{\{\ell\}}}\right)^m [L(\boldsymbol{\lambda}^0) - L^\star],$$

where σ is the smoothness constant of L, μ is a constant depending[7] only on $\boldsymbol{A}, \boldsymbol{B}$ and ρ, and $\alpha_{\{\ell\}}$ appears in Lemma 2.1.

The proof of Theorem 2.1 is provided in Sections 2.A.2 and 2.A.3. We make a few remarks regarding Theorem 2.1.

Remark 2.1. *If we used PGD on the dual (D), the linear convergence rate parameter [175, 129] would be $(1 - \mu/\sigma)$. The parameter in Theorem 2.1 has an additional factor $1/\alpha_{\{\ell\}}$, which arises due to the use of an active set method in conjunction with the augmentation rules. We present an example in Section 2.B.1 showing that this factor is perhaps unavoidable.*

Remark 2.2. *Algorithm 2.1 allows for both exact and inexact optimization of the reduced problem—the guarantees for Theorem 2.1 apply to both variants. In particular, if we use a fixed number of PGD iterations for every sub-problem, we would obtain a linear convergence rate on the total number of inner iterations. Furthermore, Theorem 2.1 applies to both deterministic and randomized augmentation rules.*

Although our theory is based on a proximal gradient update on the reduced problem, the theory also applies to other descent algorithms that have a sufficient decrease condition, e.g. accelerated proximal gradient (APG) methods [19, 175] and L-BFGS [152] with some modifications–see discussions in Remark 2.4.

Remark 2.3. *Theorem 2.1 implies that we need at most $O\left((\sigma\alpha_{\{\ell\}}/\mu)\log(1/\epsilon)\right)$-many outer iterations, to achieve an accuracy level of $\epsilon > 0$. This quantifies the worst-case convergence rate via $\alpha_{\{\ell\}}$. We note that the constant $\sigma\alpha_{\{\ell\}}/\mu$ can be large when n is large, making the speed of linear convergence slow. However, by following the arguments in the proof, one can consider a more optimistic upper bound on the number of iterations, by replacing $\alpha_{\{\ell\}}$ with $\beta_{\{\ell\}}$ (cf Lemma 2.1), as we discuss below.*

[7] See Section 2.A.2 for a characterization of μ.

Understanding the costs of different rules: To better understand the computational costs of the different rules, we consider a setting where the maximum number of selected pairs (i.e., $|\Delta'_{\{\ell\}}|$) is set to be $O(n)$ across all rules[8].

According to Remark 2.3, to achieve ϵ accuracy, we need $O((\sigma\alpha_{\{\ell\}}/\mu)\log(1/\epsilon))$ outer iterations in the worst case and $O((\sigma\beta_{\{\ell\}}/\mu)\log(1/\epsilon))$ outer iterations in the best case. Hence, for a fixed $\epsilon > 0$, the total augmentation cost is proportional to $\alpha_{\{\ell\}} \times$ *Augmentation Cost* for the worst case and $\beta_{\{\ell\}} \times$ *Augmentation Cost* for the best case. See Table 2.2 for an illustration.

Table 2.2: Comparing Augmentation Rules: We present an instance of Table 2.1 where $|\Delta'_{\{\ell\}}|$ is the same across ℓ. Specifically, we set $P = 1$ for Rule 1, $K = n$ for Rule 2, $P = 1$ for Rule 3, $M = n\sqrt{n}, K = n$ for Rule 4, and $G = \sqrt{n}, P = \sqrt{n}$ for Rule 5. Here, we ignore log-terms in the big O notation.

Rule	$\lvert\Delta'_{\{\ell\}}\rvert$	Augmentation Cost	$\alpha_{\{\ell\}}$	$\beta_{\{\ell\}}$
1	$O(n)$	$O(n^2d + d\lvert W\rvert)$	$O(n)$	$O(1)$
2	$O(n)$	$O(nd + d\lvert W\rvert)$	$O(n)$	$O(n)$
3	$O(n)$	$O(nd + d\lvert W\rvert)$	$O(n)$	$O(n)$
4	$O(n)$	$O(n\sqrt{n}d + d\lvert W\rvert)$	$O(n)$	$O(\sqrt{n})$
5	$O(n)$	$O(n\sqrt{n}d + d\lvert W\rvert)$	$O(n)$	$O(\sqrt{n})$

We make a note of the following observations from Table 2.2.

- As the maximum size of the augmentation set (i.e, $|\Delta'_{\{\ell\}}|$) at each iteration is the same across all rules, $\alpha_{\{\ell\}}$ is the same across all the five rules ℓ.
- In the worst case, different rules take the same number of iterations to achieve ϵ accuracy. The largest total augmentation costs (i.e., $\alpha_{\{\ell\}} \times$Augmentation Cost) are incurred by the purely greedy rule (Rule 1), and then the random-then-greedy rules (Rule 4 and Rule 5).
- In the best case, the greedy rules take fewer iterations—the purely greedy rule (Rule 1) is better than random-then-greedy rules (Rules 4 and 5). The purely random rules (Rules 2 and 3) have similar iteration counts for the best and worst cases (as $\alpha_{\{\ell\}} = \beta_{\{\ell\}}$).

The above discussion pertains to our theoretical guarantees. We note that the parameter $\mu/(\sigma\alpha_{\{\ell\}})$ appearing in Theorem 2.1 can be difficult to compute (see Sec-

[8]The choice of P, K, M, G are specified in the caption of Table 2.2

tion 2.A.2 for a description of μ), and may be small for large datasets making the linear rate slow in the worst-case scenario. Our numerical experiments offer a refined understanding of the operating characteristics of the different algorithms, and what occurs in practice.

We note that though our work focuses on a problem arising from convex regression, our algorithmic framework with randomized and random-then-greedy augmentation rules can also be applied to other convex quadratic programming problems involving a large number of decision variables with nonnegativity constraints.

Related work: As discussed earlier, Algorithm 2.1 (with exact optimization for the subproblems) is inspired by constraint generation, a classic tool for solving large scale linear programs [64, 34]. Recently [31] explore constraint generation for convex regression, but our framework differs. [31] use a primal approach for (2.2) and solve the restricted primal problem to *optimality* using commercial QP solvers (e.g., Gurobi). They do not present computational guarantees for their procedure. In this chapter we consider the subgradient regularized problem (P), and our algorithms are on the *dual* problem (D). We extend the framework of [31] to allow for *both* exact and inexact optimization of the reduced problems. As a main computational bottleneck in the constraint generation procedure is in the augmentation step which requires scanning all the $O(n^2)$ constraints, we present randomized augmentation rules, which examine only a small subset of the constraints, before augmenting the active-set. Our computational guarantees, which to our knowledge are novel, apply to both the exact and inexact sub-problem optimization processes and deterministic/randomized augmentation rules.

As far as we can tell, with the exception of [31], convex optimization-based approaches for convex regression that precede our work (e.g, [158, 150, 11]), do not use active set-type methods and hence apply to smaller problem instances $n \approx 3000$. We note that active-set type methods are commonly used for convex QPs see for e.g., [142, 89] and references therein. We also explore active-set type methods, (randomized) augmentation rules, inexact optimization of sub-problems, and their computa-

tional guarantees for subgradient regularized convex regression.

The primal problem (P) can also be viewed as a projection problem onto a polyhedron with $O(n^2)$ constraints. [106] consider the problem of projecting a point into a polyhedron. To solve this problem, they explore some active set ideas and develop convergence guarantees for their procedure. The active-set subproblem considered in [106] is different from what we consider. Furthermore, our work differs in using randomized augmentation rules, inexact optimization of the subproblems, and their associated computational guarantees. Using problem structure, we can address instances larger than those studied in [106].

2.4 Primal Feasibility and Duality Gap

Given a feasible solution for the full dual (D), we show how it can be used to obtain a feasible solution for the primal problem (P) while also achieving a good primal objective value (note that (2.4) need *not* deliver a primal feasible solution). A primal feasible solution is useful as it leads to a convex function estimate and also a duality gap certificate.

Given a primal candidate $(\boldsymbol{\phi}, \boldsymbol{\xi})$, we will construct $(\tilde{\boldsymbol{\phi}}, \tilde{\boldsymbol{\xi}})$ that is feasible for (P). To this end, let $v_{(i,j)} = \phi_j - \phi_i - \langle \boldsymbol{x}_j - \boldsymbol{x}_i, \boldsymbol{\xi}_i \rangle$, for all $i \neq j$ and we use the convention $v_{(j,j)} = 0$. Furthermore, let

$$\nu_j = \min_i \; v_{(i,j)} \quad \text{and} \quad \kappa_j \in \arg\min\{\|\boldsymbol{\xi}_k\| : k \in \arg\min_i v_{(i,j)}\} \tag{2.8}$$

and define $(\tilde{\boldsymbol{\phi}}, \tilde{\boldsymbol{\xi}}) := \boldsymbol{\psi}(\boldsymbol{\phi}, \boldsymbol{\xi})$ as follows

$$\tilde{\boldsymbol{\xi}}_j = \boldsymbol{\xi}_{\kappa_j} \quad \text{and} \quad \tilde{\boldsymbol{\phi}} = \boldsymbol{\phi} - \boldsymbol{\nu} + c\mathbf{1} \tag{2.9}$$

where, c is such that $\mathbf{1}^\top \tilde{\boldsymbol{\phi}} = \mathbf{1}^\top \boldsymbol{y}$. The following proposition shows that $(\tilde{\boldsymbol{\phi}}, \tilde{\boldsymbol{\xi}})$ is feasible for the full primal problem.

Proposition 2.2. *For any pair $(\boldsymbol{\phi}, \boldsymbol{\xi})$, the pair $(\tilde{\boldsymbol{\phi}}, \tilde{\boldsymbol{\xi}})$ defined in (2.9), is feasible for (P). Furthermore, if $(\boldsymbol{\phi}^\star, \boldsymbol{\xi}^\star)$ is an optimal solution for (P) then it is a fixed*

point for the map $\boldsymbol{\psi}(\cdot,\cdot)$*, i.e.,* $\boldsymbol{\psi}(\boldsymbol{\phi}^\star, \boldsymbol{\xi}^\star) = (\boldsymbol{\phi}^\star, \boldsymbol{\xi}^\star)$.

Proof of Proposition 2.2. Given $\{\phi_i\}$, $\{\boldsymbol{\xi}_i\}$ and a scalar c, we define the following piecewise linear convex function (in $\boldsymbol{x}$):

$$\varphi(\boldsymbol{x}) = \max_{i\in[n]} \ \{\langle \boldsymbol{x}, \boldsymbol{\xi}_i\rangle + (\phi_i - \langle \boldsymbol{x}_i, \boldsymbol{\xi}_i\rangle)\} + c.$$

For any j, we have

$$\max_{i\in[n]}\{\langle \boldsymbol{x}_j, \boldsymbol{\xi}_i\rangle + (\phi_i - \langle \boldsymbol{x}_i, \boldsymbol{\xi}_i\rangle)\} = \max_{i\in[n]}\{\phi_j - (\phi_j - \phi_i - \langle \boldsymbol{x}_j - \boldsymbol{x}_i, \boldsymbol{\xi}_i\rangle)\} = \phi_j - \min_{i\in[n]} v_{(i,j)}.$$

By definition of $\tilde{\boldsymbol{\phi}}, \tilde{\boldsymbol{\xi}}$ given by (2.9), it follows that

$$\varphi(\boldsymbol{x}_j) = \tilde{\phi}_j \quad \text{and} \quad \tilde{\boldsymbol{\xi}}_j = \boldsymbol{\xi}_{\kappa_j} \in \partial\varphi(\boldsymbol{x}_j).$$

Therefore, $(\tilde{\boldsymbol{\phi}}, \tilde{\boldsymbol{\xi}})$ is feasible for the full primal problem.

Due to the feasibility of $(\boldsymbol{\phi}^\star, \boldsymbol{\xi}^\star)$, we have $\boldsymbol{v}^\star = \boldsymbol{A}\boldsymbol{\phi}^\star + \boldsymbol{B}\boldsymbol{\xi}^\star \geq \mathbf{0}$, and thus $\min_i v_{(i,j)} = 0 = v_{(j,j)}$. This means $\boldsymbol{\nu} = \mathbf{0}$, so it suffices to show that $c = 0$. Note that the first term in the primal objective $\frac{1}{2}\|\boldsymbol{\phi} - \boldsymbol{y}\|^2$ is minimized when $\boldsymbol{\phi}$ and $\boldsymbol{y}$ have the same mean (otherwise we can add a constant to $\boldsymbol{\phi}$ to decrease the objective). Therefore, the c that satisfies $\mathbf{1}^\top(\boldsymbol{\phi}^\star + c\mathbf{1}) = \mathbf{1}^\top\boldsymbol{y}$ must be 0. Thus, we have shown that $\tilde{\boldsymbol{\phi}}^\star = \boldsymbol{\phi}^\star$. If $\tilde{\boldsymbol{\xi}}^\star \neq \boldsymbol{\xi}^\star$, then there exists j such that $\kappa_j \neq j$, which means $\|\boldsymbol{\xi}^\star_{\kappa_j}\| \leq \|\boldsymbol{\xi}^\star_j\|$, and we have $f(\tilde{\boldsymbol{\phi}}^\star, \tilde{\boldsymbol{\xi}}^\star) \leq f(\boldsymbol{\phi}^\star, \boldsymbol{\xi}^\star)$. This contradicts the strict minimality of $(\boldsymbol{\phi}^\star, \boldsymbol{\xi}^\star)$ as the primal objective function f is strictly convex. □

As discussed in Section 2.3.2, each $(\phi_i, \boldsymbol{\xi}_i)$ defines a hyperplane $H_i : y = \langle \boldsymbol{x} - \boldsymbol{x}_i, \boldsymbol{\xi}_i\rangle + \phi_i$ containing $P_i = (\boldsymbol{x}_i, \phi_i)$. However, $(\boldsymbol{\phi}, \boldsymbol{\xi})$ may not satisfy the convexity constraint (i.e., P_j may not lie above H_i); and $|\nu_j|$ quantifies how far P_j lies below the piecewise maximum of H_is. Intuitively, $(\tilde{\boldsymbol{\phi}}, \tilde{\boldsymbol{\xi}})$ attains feasibility by taking a piecewise maximum of these hyperplanes and then adjusts itself by a constant in an attempt to decrease the primal objective.

Duality Gap: In light of strong duality between (P) and (D), we have $L(\boldsymbol{\lambda}^\star) =$

$-f(\boldsymbol{\phi}^\star, \boldsymbol{\xi}^\star)$. From a dual feasible solution $\boldsymbol{\lambda}$, we can obtain a primal variable $(\boldsymbol{\phi}, \boldsymbol{\xi})$ via (2.4). If $\boldsymbol{\psi}(\boldsymbol{\phi}, \boldsymbol{\xi})$ is a primal *feasible* solution obtained from $(\boldsymbol{\phi}, \boldsymbol{\xi})$ by the operation defined in (2.9), then we can compute a duality gap via:

$$\underline{L}(\boldsymbol{\lambda}) = -f(\boldsymbol{\psi}(\boldsymbol{\phi}, \boldsymbol{\xi})) \leq L^\star \leq L(\boldsymbol{\lambda}).$$

The gap $L(\boldsymbol{\lambda}) - \underline{L}(\boldsymbol{\lambda})$ equals zero if and only if $\boldsymbol{\lambda}$ is an optimal solution for (D).

2.5 Numerical Experiments

We present numerical experiments to study the different variants of our algorithm; and compare it with current approaches. As our focus is on large-scale instances of (P_0), we study both real and synthetic datasets in the range $n \sim 10^4 - 10^5$ and $4 \leq d \leq 20$.

Datasets: We consider the following synthetic and real datasets for the experiments.

Synthetic Data: Following [158], we generate data via the model $y_i = \phi^0(\boldsymbol{x}_i) + \epsilon_i, i \in [n]$ where, $\phi^0(\boldsymbol{x})$ is a convex function, $\epsilon_i \overset{\text{iid}}{\sim} N(0, \gamma^2)$ and γ is chosen to match a specified value of signal-to-noise ratio, SNR $= \|\boldsymbol{\phi}^0\|^2/\|\boldsymbol{\epsilon}\|^2 = 3$. The covariates are drawn independently from a uniform distribution on $[-1, 1]^d$. Every feature is normalized to have zero mean and unit ℓ_2-norm; and $\boldsymbol{y}$ has zero mean and unit ℓ_2-norm. We consider the following choices of the underlying true convex function $\phi^0(\boldsymbol{x})$.

- SD1: $\phi^0(\boldsymbol{x}) = \|\boldsymbol{x}\|_2^2$
- SD2: $\phi^0(\boldsymbol{x}) = \max_{1 \leq i \leq 2d}\{\boldsymbol{\xi}_i^\top \boldsymbol{x}\}$, where $\boldsymbol{\xi}_i \in \mathbb{R}^d$ are independently drawn from the uniform distribution $[-1, 1]^d$.

Real Data: We also consider the following real datasets in our experiments:

- RD1, RD2: These two datasets, studied in [130] have $n = 10,000$ and $d = 4$.
- RD3, RD4: These two datasets, studied in [229] have $n = 10,000$ and $d = 4$.
- RD5: This dataset, studied in [148], has $n = 10,000$ and $d = 4$.
- RD6: This dataset, studied in [131, 208], has $n = 5,000$ and $d = 4$.

- RD7: This dataset, from earlier work [164, 158, 31] has $n = 30,000$ and $d = 4$.
- RD8: This dataset, from [186, 112, 14], has $n = 10,000$ and $d = 4$.

For additional details on the real datasets see Section 2.C.1. In all above datasets, covariates and response are centered and scaled so that each variable has unit ℓ_2-norm.

Algorithms: We compare our approach versus the cutting plane (CP) method [31] and the ADMM method [158] using the authors' implementations. All algorithms are run with the time limit of 12 hours[9]. For our algorithms, we consider two-stage methods: In the first stage, we use random rules 2/4 for augmentation and perform inexact optimization over the active set — empirically, this results in good initial progress in the objective value but then the progress slows down as the augmentation rules include very few violated constraints. When the number of added constraints is less than $0.005n$ for consecutive 5 iterations[10] we switch to the second stage where we use occasional greedy rules 1/5 for augmentation and exact optimization of sub-problems. Note that our theory (Section 2.3) guarantees convergence of this two-stage procedure. Additional details on algorithm parameter choices can be found in Section 2.C.2. In this section, we will use Rule a-b ($a \in \{2, 4\}, b \in \{1, 5\}$) to denote different variants of our algorithm with random rule a (Stage 1) and greedy rule b (Stage 2).

Software Specifications: All computations were carried out on MIT's Engaging Cluster on an Intel Xeon 2.30GHz machine, with one CPU and 8GB of RAM. For ADMM and CP, we used a larger amount of memory 64GB RAM. Our algorithms are written in Julia (v1.5), and our code is available on github at:

`https://github.com/wenyuC94/ConvexRegression`

Performance of proposed algorithms: Figure 2-1 presents the relative objective of the dual, defined as

$$\text{Rel. Obj.} = (L(\boldsymbol{\lambda}^t) - L^\star)/(|L^\star| + 1)$$

[9]For the cutting plane method which uses Gurobi, it can take a long time to solve the reduced sub-problem thereby exceeding the allocated 12 hr time-limit before obtaining an accurate solution.

[10]This is a choice we used in our experiments, and can be tuned in general for performance benefits.

as well as the primal infeasibility (`pinfeas`)—also used in [158, 31]—defined as[11]

$$\texttt{pinfeas} = \frac{1}{n}\|(\nabla L(\boldsymbol{\lambda}^t))_+\|$$

for different algorithms versus time (in seconds). Above, $L^\star$ is taken as the minimum objective among solutions obtained by all the algorithms running for 12 hours. Figure 2-1 considers synthetic datasets SD1 and SD2 with different n's and d's, and the real dataset RD1. We note that the choices of ρ displayed in Figure 2-1 correspond to good statistical performance—this is discussed in further detail below and in Figure 2-2. Figure 2-1 suggests that our algorithms perform better than CP and ADMM – both in terms of Rel. Obj. and primal infeasibility. Note that ADMM [158] does not use active-set methods and consumes prohibitively large memory when $n \geq 30{,}000$.

Table 2.3 presents a snapshot of the runtimes of our proposed algorithms, CP and ADMM. Here, the runtime corresponds to the time (s) taken by an algorithm to achieve a 0.05 relative objective. Our proposed two-stage algorithms can achieve this accuracy quite quickly, while CP and ADMM are often unable to converge to that accuracy. We observe that for our algorithms, runtime generally increases with smaller ρ-values. Table 2.4 presents results for larger datasets with $n = 100{,}000$—we show two of our methods and do not include greedy augmentation Rule 1 due to large computational costs. As expected, for such large problems randomized augmentation rules play a key role.

Subgradient regularization can improve statistical performance: As mentioned earlier, the presence of ℓ_2-regularization on the subgradients (i.e., a value of $\rho > 0$) in (P_0), can lead to improved statistical performance of the convex function estimate when compared to the unregularized case (i.e., when $\rho = 0$). Intuitively, this is due in part to the behavior of the convex regression fit near the boundary of the convex hull of the covariates [13, 158]. The ℓ_2-regularization on subgradients can help regulate boundary behavior and improve performance of the estimator: see [158] for

[11] For CP [31] and ADMM [158], Rel. Obj. is defined as $|f(\boldsymbol{\phi}^t, \boldsymbol{\xi}^t) - f^\star|/(1+|f^\star|)$, where $f^\star = -L^\star$; `pinfeas` is defined as $\|(\boldsymbol{A}\boldsymbol{\phi}^t + \boldsymbol{B}\boldsymbol{\xi}^t)_-\|/n$. Here $\boldsymbol{a}_+$ and $\boldsymbol{a}_-$ denotes the vector of $[\max\{a_j, 0\}]_j$ and $[\min\{a_j, 0\}]_j$, respectively.

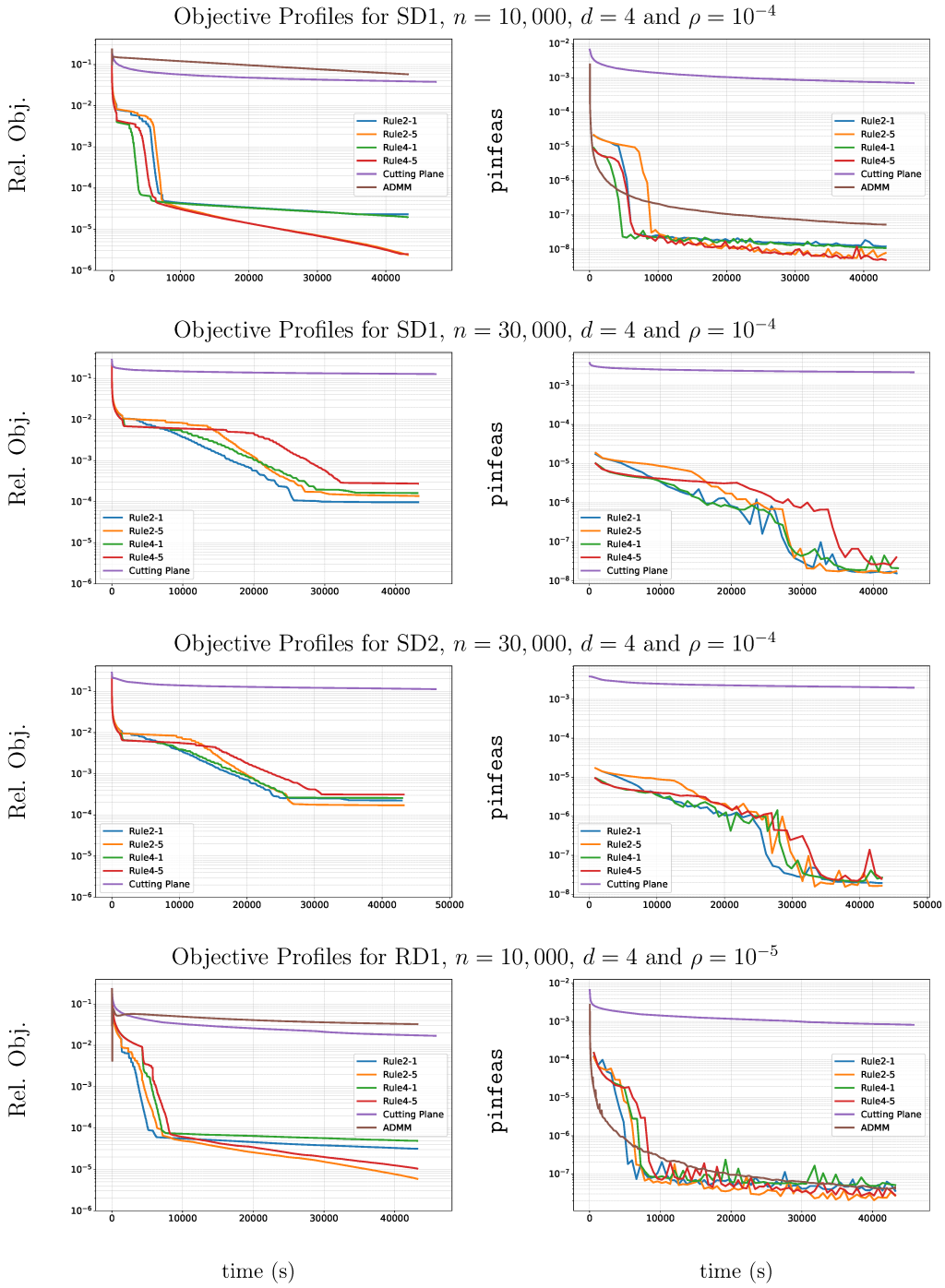

Figure 2-1: Plots (in log-scale) of Relative Objective [left panel] and primal infeasibility [right panel] versus time (secs). We consider three synthetic data sets (top 3 rows) and a real dataset (bottom row). We compare our algorithms against the cutting plane method [31] and the ADMM method [158]. For each algorithm, we run 5 repetitions, each bold line corresponds to the median of the profiles of one algorithm. The ADMM profiles (2nd, 3rd rows) are missing as they run out of memory (64GB).

Table 2.3: Comparison of runtime (s) of our algorithms versus CP [31] and ADMM [158]. Here runtime refers to the time taken to achieve a Rel. Obj. of 5×10^{-2}. We report the median runtime and standard error (in bracket) across 5 replications (random instances). Note: '-' means that no replication of the algorithm achieves this level of relative accuracy within 4hrs, '-*' means that some replications encountered convergence issues and others did not reach the tolerance within 4hrs, '**' means all replications crash due to either numerical/memory problems. The entry 4320.90* (column=CP and row=RD4) means that some of replications crashed due to numerical/memory issues, and we report the median runtime for the replications that did not crash.

data	$\frac{n}{10^4}$	d	$\log_{10}\rho$	Rule2-1	Rule2-5	Rule4-1	Rule4-5	CP [31]	ADMM [158]
SD1	3	4	-3	3.81 (0.95)	3.67 (0.66)	2.33 (0.25)	2.41 (0.24)	-	**
SD1	3	4	-4	52.66 (1.93)	52.35 (35.44)	44.14 (9.56)	43.71 (45.87)	-	**
SD1	3	10	-3	31.56 (10.62)	21.06 (12.57)	10.64 (8.69)	11.50 (2.25)	-	**
SD1	3	10	-4	1290.51 (114.15)	1289.37 (355.51)	364.57 (202.95)	362.25 (192.42)	-*	**
SD1	3	20	-3	101.14 (72.88)	180.22 (64.26)	95.69 (25.63)	93.91 (28.92)	-	**
SD1	3	20	-4	2767.88 (373.64)	2774.59 (416.69)	939.66 (222.82)	898.22 (297.24)	-	**
SD2	3	4	-3	4.29 (3.22)	3.51 (0.32)	2.33 (0.14)	2.52 (1.32)	-	**
SD2	3	4	-4	39.22 (1.83)	44.87 (21.84)	37.05 (21.41)	37.14 (21.96)	-	**
SD2	3	10	-3	18.76 (12.56)	19.00 (3.86)	10.60 (0.77)	14.00 (8.59)	-	**
SD2	3	10	-4	975.48 (344.82)	978.28 (330.26)	404.70 (223.14)	403.52 (217.71)	-*	**
SD2	3	20	-3	124.51 (9.55)	124.22 (9.39)	66.80 (49.30)	66.01 (49.05)	-	**
SD2	3	20	-4	4292.47 (365.84)	4693.06 (434.83)	1400.07 (249.20)	1187.88 (119.68)	-	**
RD1	1	4	-4	28.74 (2.72)	28.68 (2.81)	22.16 (3.96)	21.72 (4.12)	6827.02 (963.13)	-
RD1	1	4	-5	168.99 (13.11)	161.28 (17.96)	276.40 (20.29)	273.31 (20.43)	3704.07 (876.34)	9729.61 (609.72)
RD2	1	4	-4	47.16 (4.77)	61.88 (3.99)	24.19 (1.81)	23.02 (2.65)	-	
RD2	1	4	-5	264.68 (21.88)	255.80 (21.92)	115.46 (43.57)	117.49 (41.41)	**	248.71 (9.77)
RD3	1	4	-4	18.16 (2.07)	18.01 (2.06)	11.09 (2.04)	11.31 (2.09)	-	-
RD3	1	4	-5	83.41 (4.48)	70.47 (6.94)	75.07 (6.94)	74.31 (6.97)	7805.34 (1325.51)	244.73 (4.75)
RD4	1	4	-4	17.59 (0.76)	18.04 (0.85)	17.72 (1.08)	15.92 (1.19)	-	-
RD4	1	4	-5	128.86 (4.95)	128.51 (5.00)	107.64 (6.75)	122.39 (5.62)	4320.90*	261.39 (8.66)
RD5	1	4	-4	11.98 (0.82)	11.76 (0.84)	7.72 (0.53)	7.70 (0.55)	-	-
RD5	1	4	-5	104.33 (7.92)	114.19 (9.29)	83.22 (5.31)	81.75 (2.51)	-	248.79 (7.82)
RD6	$\frac{1}{2}$	4	-4	3.59 (0.22)	3.15 (0.27)	2.61 (0.25)	2.34 (0.26)	233.47 (6.98)	1707.11 (115.08)
RD6	$\frac{1}{2}$	4	-5	9.35 (0.68)	10.81 (0.53)	7.41 (1.05)	6.92 (1.10)	64.06 (13.64)	71.84 (5.13)
RD7	3	4	-4	20.84 (3.08)	20.84 (3.13)	17.15 (3.35)	25.51 (3.65)	-	**
RD7	3	4	-5	40.81 (7.80)	41.31 (7.73)	39.99 (16.76)	85.39 (17.16)	-	**
RD8	1	4	-4	4.87 (0.12)	5.25 (0.46)	4.98 (0.85)	5.00 (0.86)	-	-
RD8	1	4	-5	37.56 (4.07)	33.31 (4.28)	54.85 (8.00)	55.22 (9.21)	-	235.58 (5.29)

theoretical support. Here, we present some numerical evidence to support this observation. Figure 2-2 presents the training and test root mean squared error (RMSE)[12] on some real datasets. To quantify the performance of the convex function estimate near the boundary of the convex hull, we compute the RMSE on the boundary points[13]. Figure 2-2 shows what values of $\rho > 0$ result in good statistical performance

[12]The RMSE is computed by the following procedure: (i) obtain the primal feasible solution $(\tilde{\boldsymbol{\phi}}, \tilde{\boldsymbol{\xi}})$ according to Section 2.4, (ii) obtain the prediction $\hat{y}$ for each data point $\boldsymbol{x}$ via a piecewise-maximum interpolation scheme $\hat{y} = \max_i\{\tilde{\phi}_i + \langle\tilde{\xi}_i, \boldsymbol{x} - \boldsymbol{x}_i\rangle\}$; (iii) evaluate RMSE based on the predictions $\hat{y}$ and the observed values y.

[13]We compute the convex hull of the training set and identify points in the test set near the boundary of this convex hull, according to the distance of each point to the convex hull.

Table 2.4: Runtime (s) of our algorithms, and the cutting plane (CP) method [31] for $n = 100{,}000$. ADMM runs out of memory (64GB) on these instances. See Table 2.3 for more details on the notations.

data	n	d	$\log_{10}\rho$	Rule2-5	Rule4-5	CP [31]
SD1	100000	4	-3	19.15 (5.05)	16.50 (1.12)	-
SD1	100000	4	-4	94.52 (21.63)	102.66 (23.53)	-
SD1	100000	10	-3	27.30 (7.66)	22.48 (6.07)	-
SD1	100000	10	-4	598.83 (54.45)	627.96 (59.55)	-
SD1	100000	20	-3	71.08 (12.05)	38.84 (6.29)	-
SD1	100000	20	-4	2428.77 (232.50)	1041.10 (113.51)	-*
SD2	100000	4	-3	29.89 (2.62)	28.46 (5.69)	-
SD2	100000	4	-4	121.77 (23.12)	111.20 (19.85)	-
SD2	100000	10	-3	27.08 (2.60)	28.15 (3.17)	-
SD2	100000	10	-4	509.58 (27.91)	422.83 (73.77)	-
SD2	100000	20	-3	75.59 (13.97)	49.22 (6.37)	-
SD2	100000	20	-4	2810.69 (262.19)	1397.47 (134.45)	-

(in terms of RMSE). In particular, values of $\rho = 10^{-4} - 10^{-5}$ generally result in good RMSE performance for the real data sets. We also observe that $\rho = 10^{-3} - 10^{-4}$ result in good statistical performance for the synthetic data sets. In practice, we recommend selecting a value of ρ that minimizes RMSE on a validation dataset (or based on cross-validation).

2.6 Conclusion

We present large-scale algorithms for subgradient regularized multivariate convex regression, a problem of key importance in nonparametric regression with shape constraints. We present an active set type method on the smooth dual problem (D): we allow for inexact optimization of the reduced sub-problems and use randomized rules for augmenting the current active set. We establish novel computational guarantees for our proposed algorithms. For large-scale instances, our approach appears to be more suited to obtain low-moderate accuracy solutions. Exploiting problem-structure, our open source toolkit can deliver approximate solutions for instances with $n \approx 10^5$ and $d = 10$ (a QP with 10 billion decision variables) within a few minutes on a modest computer. Our approach appears to work well as long as the tuning parameter ρ is not too small, while still corresponding to good statistical models. For

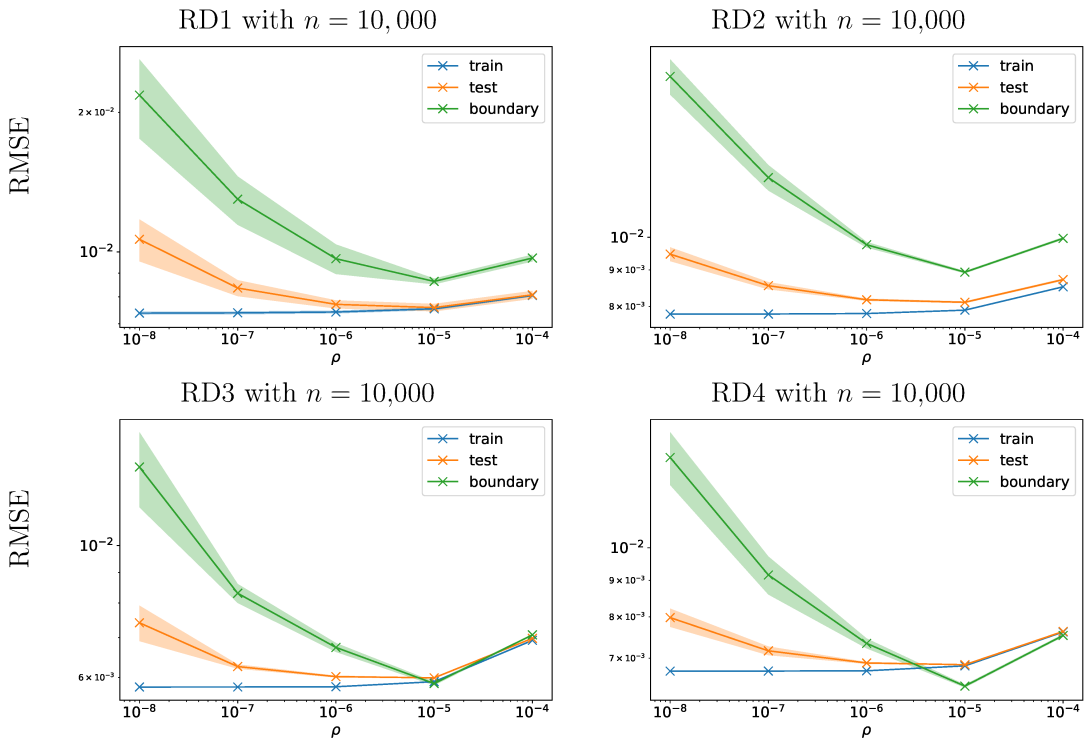

Figure 2-2: Plots (in log-scale) of RMSEs evaluated on training set, test set and boundary test set versus ρ's for 4 real datasets. We consider ten replications (subsamples) and plot the mean (markers) and standard error (error bars). The training RMSE decreases with ρ and appears to stabilize when ρ becomes smaller than 10^{-5} (approx). We observe that the minimum RMSE on the test/boundary set occurs when $\rho \approx 10^{-5}$, and this value is quite close to the test RMSE at $\rho \approx 10^{-4}$. We study both these ρ-values in our runtime comparisons.

solving (P) with $\rho = 0$ (or numerically very close to zero), our approach would not apply and we recommend using the cutting plane procedure of [31] though this does not have computational guarantees.

2.A Appendix: Proofs

In this section, we present the proofs of Lemma 2.1 and Theorem 2.1. The proof of Theorem 2.1 is based on Lemmas 2.1 and 2.4. We first present the proof of Lemma 2.1 (Section 2.A.1); then present Lemma 2.4 (Section 2.A.2), followed by the proof of Theorem 2.1.

2.A.1 Proof of Lemma 2.1

For any $\boldsymbol{u} \in \mathbb{R}^N$ and $P \leq N$, we define two norms:

$$\|\boldsymbol{u}\|_{\mathcal{G}[P]} = \left(\sum_{i \in \mathcal{G}[[N],P,\boldsymbol{u}]} u_i^2 \right)^{1/2} \quad \text{and} \quad \|\boldsymbol{u}\|_{\mathcal{U}[P]} = \mathbb{E} \left(\sum_{i \in \mathcal{U}[[N],P]} u_i^2 \right)^{1/2} \tag{2.10}$$

where, $\mathcal{G}[[N], P, \boldsymbol{u}], \mathcal{U}[P]$ are defined in Definition 2.2 and $\mathbb{E}(\cdot)$ is the expectation wrt scheme $\mathcal{U}$. Proposition 2.3 links these norms to the Euclidean norm:

Proposition 2.3. *For $\boldsymbol{u} \in \mathbb{R}^N$ and the norms defined in* (2.10)*, the following statement holds:*

$$\|\boldsymbol{u}\|^2 \geq \|\boldsymbol{u}\|^2_{\mathcal{G}[P]} \geq \frac{P}{N}\|\boldsymbol{u}\|^2 = \|\boldsymbol{u}\|^2_{\mathcal{U}[P]}.$$

Proof of Proposition 2.3. For $\boldsymbol{u} \in \mathbb{R}^N$, we see that $\|\boldsymbol{u}\|^2_{\mathcal{U}[P]} = (P/N)\|\boldsymbol{u}\|^2$. Notice that

$$\|\boldsymbol{u}\|^2_{\mathcal{G}[P]} \;=\; \max_{\boldsymbol{\pi}} \sum_{\omega \in [N]} \pi_\omega |u_\omega|^2 \quad \text{s.t.} \quad \sum_{\omega \in [N]} \pi_\omega \leq P, \quad 0 \leq \pi_\omega \leq 1, \forall \omega.$$

Since $\pi_\omega = P/N \;\forall \omega$, is feasible for the above problem, it follows that

$$\|\boldsymbol{u}\|^2_{\mathcal{G}[P]} \geq \frac{P}{N}\|\boldsymbol{u}\|^2 = \|\boldsymbol{u}\|^2_{\mathcal{U}[P]}.$$

Equality above is attained if and only if $u_\omega = C \;\forall \omega$ for some C. Furthermore, we note that

$$\|\boldsymbol{u}\|^2_{\mathcal{G}[P]} = \sum_{\omega \in \mathcal{G}[[N],P]} |u_\omega|^2 \leq \sum_{\omega \in [N]} |u_\omega|^2 = \|\boldsymbol{u}\|^2,$$

and this equality is attained if and only if $u_\omega = 0$ for all $\omega \notin \mathcal{G}[[N], P]$, i.e. the $N - P$ smallest values of $|u_\omega|$ are 0. □

Proof of Lemma 2.1. We divide the proof into 5 parts depending upon the 5 rules.

Rule 1: Greedy within each Block. $(\delta_1(\boldsymbol{\theta}, \{\Omega_i\}) = \cup_{i=1}^n \mathcal{G}[\Omega_i, P, \boldsymbol{\theta}]$, $\alpha_{\{1\}} = \frac{n-1}{P}$,

and $\beta_{\{1\}} = 1$.) For this selection rule, we have

$$\|\boldsymbol{\theta}\|_{\{1\}}^2 = \sum_{i=1}^{n} \|\boldsymbol{\theta}_{\Omega_i}\|_{\mathcal{G}[P]}^2.$$

It is easy to see that $\|\boldsymbol{\theta}\|_{\{1\}}$ is a norm. It follows from Proposition 2.3 that

$$\|\boldsymbol{\theta}\|^2 = \sum_{i=1}^{n} \|\boldsymbol{\theta}_{\Omega_i}\|^2 \geq \sum_{i=1}^{n} \|\boldsymbol{\theta}_{\Omega_i}\|_{\mathcal{G}[P]}^2 \geq \sum_{i=1}^{n} \frac{P}{n-1} \|\boldsymbol{\theta}_{\Omega_i}\|^2 = \frac{P}{n-1} \|\boldsymbol{\theta}\|^2.$$

Therefore, we have $\alpha_{\{1\}} = \frac{n-1}{P}$, and $\beta_{\{1\}} = 1$.

Rule 2: Random. ($\delta_2(\boldsymbol{\theta}, \{\Omega_i\}) = \mathcal{U}[\Omega, K]$, $\alpha_{\{2\}} = \frac{n(n-1)}{K}$ and $\beta_{\{2\}} = \frac{n(n-1)}{K}$).

For this selection rule, we have

$$\|\boldsymbol{\theta}\|_{\{2\}}^2 = \|\boldsymbol{\theta}\|_{\mathcal{U}[K]}^2 = \frac{K}{n(n-1)} \|\boldsymbol{\theta}\|^2.$$

Thus, in this case the norm-equivalence constants are $\alpha_{\{2\}} = \beta_{\{2\}} = \frac{n(n-1)}{K}$.

Rule 3: Random within each Block. ($\delta_3(\boldsymbol{\theta}, \{\Omega_i\}) = \cup_{i=1}^{n} \mathcal{U}[\Omega_i, P]$, $\alpha_{\{3\}} = \frac{n-1}{P}$, and $\beta_{\{3\}} = \frac{n-1}{P}$.) For this selection rule, we have

$$\|\boldsymbol{\theta}\|_{\{3\}}^2 = \sum_{i=1}^{n} \|\boldsymbol{\theta}_{\Omega_i}\|_{\mathcal{U}[P]}^2 = \sum_{i=1}^{n} \frac{P}{n-1} \|\boldsymbol{\theta}_{\Omega_i}\|^2 = \frac{P}{n-1} \|\boldsymbol{\theta}\|^2.$$

Hence, it follows that $\alpha_{\{3\}} = \beta_{\{3\}} = \frac{n-1}{P}$.

Rule 4: Random then Greedy. ($\delta_4(\boldsymbol{\theta}, \{\Omega_i\}) = \mathcal{G}[\mathcal{U}[\Omega, M], K, \boldsymbol{\theta}]$, $\alpha_{\{4\}} = \frac{n(n-1)}{K}$, and $\beta_{\{4\}} = \frac{n(n-1)}{M}$.) We adapt the proof of [154] to show that under this selection rule

$$\|\boldsymbol{\theta}\|_{\{4\}}^2 = \sum_{l=1}^{n(n-1)} \pi(l) |\theta_{(l)}|^2,$$

where $|\theta_{(l)}|$ is the l-th largest value in $\{|\theta_\omega|\}_{\omega \in \Omega}$ and $\pi(l)$ is given by

$$\pi(l) = \frac{M}{n(n-1)} \sum_{k=1}^{K} \frac{\binom{l-1}{k-1} \binom{n(n-1)-l}{M-k}}{\binom{n(n-1)-1}{M-1}}.$$

Here, by convention, we define $\binom{N}{\alpha} = 0$ if $\alpha < 0$ or $\alpha > N$.

Let $\pi(l)$ be the probability that $|\theta_{(l)}|$ is selected. Since the subsample is selected uniformly at random, it suffices to count the number of combinations that include $|\theta_{(l)}|$ and those in which $|\theta_{(l)}|$ ranks among the top K values. This is equivalent to choosing $k-1 (\leq K-1)$ elements from $\{|\theta_{(s)}|\}_{s \leq l-1}$, selecting the element $|\theta_{(l)}|$ and then choosing the remaining $(M-k)$ elements from the rest. Therefore, the number of such combinations is

$$N(l) = \sum_{k=1}^{K} \binom{l-1}{k-1} \binom{n(n-1)-l}{M-k}.$$

Thus,

$$\pi(l) = \frac{N(l)}{\binom{n(n-1)}{M}} = \frac{M}{n(n-1)} \sum_{k=1}^{K} \frac{\binom{l-1}{k-1}\binom{n(n-1)-l}{M-k}}{\binom{n(n-1)-1}{M-1}}.$$

Notice that when $l \leq K$ (i.e., $l-1 \leq K-1$), then

$$N(l) = \sum_{k=1}^{l} \binom{l-1}{k-1} \binom{n(n-1)-l}{M-k} = \binom{n(n-1)-1}{M-1},$$

and thus $\pi(l) = \frac{M}{n(n-1)}$.

When $l \geq n(n-1) - (M-K) + 1$ and $M - k \geq M - 1 - K > n(n-1) - l$, then $N(l) = 0$ and thus $\pi(l) = 0$.

As each element appears in the same number of combinations of size M (in the random selection step), and the greedy selection step favors the larger one, we have the following ordering: $\pi(1) \geq \ldots \geq \pi(n(n-1))$. Therefore, $\|\boldsymbol{\theta}\|_{\{4\}}$ is a norm.

Since $\pi(l) = \mathbb{E}[\mathbb{I}_{\{(l) \in \mathcal{G}[\mathcal{U}[\Omega, M], K]\}}]$, it follows that

$$\sum_{l=1}^{n(n-1)} \pi(l) = \mathbb{E}\left[\sum_{l=1}^{n(n-1)} \mathbb{I}_{\{(l) \in \mathcal{G}[\mathcal{U}[\Omega, M], K]\}}\right] = K.$$

Then, we have

$$\|\boldsymbol{\theta}\|_{\{4\}}^2 = \sum_{l=1}^{n(n-1)} \pi(l)|\theta_{(l)}|^2 \geq \sum_{l=1}^{n(n-1)} \frac{K}{n(n-1)}|\theta_{(l)}|^2 = \frac{K}{n(n-1)}\|\boldsymbol{\theta}\|^2. \tag{2.11}$$

On the other hand, since $\pi(l) \leq \pi(1) = \frac{M}{n(n-1)}$, it follows that

$$\|\boldsymbol{\theta}\|_{\{4\}}^2 = \sum_{l=1}^{n(n-1)} \pi(l)|\theta_{(l)}|^2 \leq \sum_{l=1}^{n(n-1)} \frac{M}{n(n-1)}|\theta_{(l)}|^2 = \frac{M}{n(n-1)}\|\boldsymbol{\theta}\|^2. \tag{2.12}$$

Hence from (2.11) and (2.12), we obtain $\alpha_{\{4\}} = \frac{n(n-1)}{K}$ and $\beta_{\{4\}} = \frac{n(n-1)}{M}$.

Rule 5: Random Blocks then Greedy within each Block.
($\delta_5(\boldsymbol{\theta}, \{\Omega_i\}) = \cup_{i\in\mathcal{U}[[n],G]}\mathcal{G}[\Omega_i, P, \boldsymbol{\theta}]$, $\alpha_{\{5\}} = \frac{n(n-1)}{GP}$, and $\beta_{\{5\}} = \frac{n}{G}$.)

By the selection rule, we have

$$\|\boldsymbol{\theta}\|_{\{5\}}^2 = \frac{G}{n}\sum_{i=1}^{n} \|\boldsymbol{\theta}_{\Omega_i}\|_{\mathcal{G}[P]}^2 = \frac{G}{n}\|\boldsymbol{\theta}\|_{\{1\}}^2$$

with $\alpha_{\{5\}} = \frac{n}{G}\alpha_{\{1\}} = \frac{n(n-1)}{GP}$ and $\beta_{\{5\}} = \frac{n}{G}\beta_{\{1\}} = \frac{n}{G}$.

□

2.A.2 Auxiliary lemmas for the proof of Theorem 2.1

Here we present Lemmas 2.3 and 2.4 that will be used for the proof of Theorem 2.1.

Lemma 2.2 (Hoffman's Lemma [122]). *Let $\Lambda = \{\boldsymbol{\lambda} : \boldsymbol{D}\boldsymbol{\lambda} = \boldsymbol{s}, \boldsymbol{\lambda} \leq \boldsymbol{0}\}$. There is a constant $\mu_{\boldsymbol{D}} > 0$ that depends only on $\boldsymbol{D}$ such that for any $\boldsymbol{\lambda} \leq \boldsymbol{0}$, there exists an $\boldsymbol{\lambda}_0 \in \Lambda$ with*

$$\|\boldsymbol{D}\boldsymbol{\lambda} - \boldsymbol{s}\|^2 \geq \mu_{\boldsymbol{D}}\|\boldsymbol{\lambda} - \boldsymbol{\lambda}_0\|^2.$$

The constant $\mu_{\boldsymbol{D}}$ is called the Hoffman's constant associated with $\boldsymbol{D}$.

Let $\boldsymbol{D}^\star = \begin{bmatrix} \boldsymbol{A}^\top \\ \boldsymbol{B}^\top/\sqrt{\rho} \end{bmatrix}$ be the coefficient matrix corresponding to set of optimal solutions $\boldsymbol{\Lambda}^\star$ with $W = \Omega$ in (2.3). We get the following sufficient descent lemma

according to [129]:

Lemma 2.3 ([129] Case 3, Page 18)**.** *For any $\boldsymbol{\lambda} \leq \mathbf{0}$ and optimal dual solution $\boldsymbol{\lambda}^\star$ of (D), we have*

$$L(\boldsymbol{\lambda}^\star) \geq L(\boldsymbol{\lambda}) - \frac{1}{2\mu_{\boldsymbol{D}^\star}} \mathcal{D}(\boldsymbol{\lambda}, \nabla L(\boldsymbol{\lambda}), \mu_{\boldsymbol{D}^\star}), \tag{2.13}$$

where $\mu_{\boldsymbol{D}^\star} > 0$ is the Hoffman's constant associated with $\boldsymbol{D}^\star$.

Definition 2.5. *Define a function $\mathcal{D}(\cdot,\cdot,\cdot) : \mathbb{R}^N_- \times \mathbb{R}^N \times \mathbb{R}_{++} \to \mathbb{R}_+$ as*

$$\mathcal{D}(\boldsymbol{\lambda}, \boldsymbol{\theta}, \mu) = -2\mu \min_{\boldsymbol{\lambda}' \leq 0} \left\{ \langle \boldsymbol{\theta}, \boldsymbol{\lambda}' - \boldsymbol{\lambda} \rangle + \frac{\mu}{2} \|\boldsymbol{\lambda}' - \boldsymbol{\lambda}\|^2 \right\}.$$

Lemma 2.4 makes use of the following useful proposition.

Proposition 2.4. *Define $d(\cdot,\cdot,\cdot) : \mathbb{R}_- \times \mathbb{R}_+ \times \mathbb{R}_{++} \to \mathbb{R}$ as*

$$d(\lambda, \theta, \mu) = -2\mu \min_{\lambda' \leq 0} \left\{ \theta(\lambda' - \lambda) + \frac{\mu}{2} (\lambda' - \lambda)^2 \right\}.$$

We have the following:

(a) if $\lambda = 0$, then for any $\mu > 0$, $d(\lambda, \theta, \mu) = \max\{\theta, 0\}^2$;

(b) if $\lambda\theta = 0$ with $\lambda \leq 0$ and $\theta \leq 0$, then for any $\mu > 0$, $d(\lambda, \theta, \mu) = 0$;

(c) if $\lambda \leq 0$, then for any $\mu > 0$, $d(\lambda, \theta, \mu) \geq 0$.

Proof. Part (a): If $\lambda = 0$, then

$$d(\lambda, \theta, \mu) = -2\mu \min_{\lambda' \leq 0} \left\{ \theta\lambda' + \frac{\mu}{2} (\lambda')^2 \right\} = \begin{cases} 0 & \text{if } \theta \leq 0 \\ \theta^2 & \text{if } \theta > 0 \end{cases},$$

i.e. $d(\lambda, \theta, \mu) = \max\{\theta, 0\}^2$.

Part (b): If $\lambda\theta = 0$ with $\lambda = 0$ and $\theta \leq 0$, then $d(\lambda, \theta, \mu) = 0$ follows from Part (a).

If $\lambda\theta = 0$ with $\lambda < 0$, then $\theta = 0$, so $d(\lambda, \theta, \mu) = -2\mu \min_{\lambda' \leq 0} \{\frac{\mu}{2}(\lambda' - \lambda)^2\} = 0$.

Part (c): If $\lambda \leq 0$, then $\lambda' = \lambda \leq 0$ is always feasible, so $d(\lambda, \theta, \mu) \geq 0$. □

Lemma 2.4. *Suppose there exists $\alpha \geq 1$ such that for any m and W^0 the following*

$$\alpha \mathbb{E}_{\eta_{m+1}} \left[\sum_{\omega \in W^{m+1} \setminus W^m} \max\{\nabla_{\lambda_\omega} L(\boldsymbol{\lambda}^m), 0\}^2 \right] \geq \sum_{\omega \in \Omega \setminus W^m} \max\{\nabla_{\lambda_\omega} L(\boldsymbol{\lambda}^m), 0\}^2 \quad (2.14)$$

holds almost surely, where $\mathbb{E}_{\eta_{m+1}}$ denotes expectation over all the random sources in $(m+1)$-th iteration (conditional on events up to iteration m). Then, the following holds:

$$\mathbb{E}_{\eta_{m+1}}[L(\boldsymbol{\lambda}^{m+1}) - L^\star] \leq \left(1 - \frac{\mu_{\boldsymbol{D}^\star}}{\alpha\sigma}\right)[L(\boldsymbol{\lambda}^m) - L^\star].$$

Proof of Lemma 2.4. We make use of the following notation in the proof:

$$\lambda_\omega^m \text{ is the } \omega\text{-th component of } \boldsymbol{\lambda}^m, \qquad \nabla_\omega^m = \nabla_{\lambda_\omega} L(\boldsymbol{\lambda}^m) \quad \text{and} \quad \tilde{\nabla}_\omega^m = \max\{\nabla_\omega^m, 0\}.$$

The proof has three steps, and we consider both exact and inexact cases in each step.

Step 1: In the first part, we will show that

$$\begin{aligned} & L(\boldsymbol{\lambda}^m) - L(\boldsymbol{\lambda}^\star) \\ \leq & \begin{cases} \dfrac{1}{2\mu_{\boldsymbol{D}^\star}} \displaystyle\sum_{\omega \in \Omega \setminus W^m} (\tilde{\nabla}_\omega^m)^2 & \text{(exact case)} \\ \dfrac{1}{2\mu_{\boldsymbol{D}^\star}} \left[\displaystyle\sum_{\omega \in W^m} d(\lambda_\omega^m, \nabla_\omega^m, \mu_{\boldsymbol{D}^\star}) + \sum_{\omega \in \Omega \setminus W^m} (\tilde{\nabla}_\omega^m)^2 \right] & \text{(inexact case).} \end{cases} \end{aligned} \quad (2.15)$$

It follows from Lemma 2.3 that

$$L(\boldsymbol{\lambda}^\star) \geq L(\boldsymbol{\lambda}^m) - \frac{1}{2\mu_{\boldsymbol{D}^\star}} \mathcal{D}(\boldsymbol{\lambda}^m, \nabla L(\boldsymbol{\lambda}^m), \mu_{\boldsymbol{D}^\star}). \quad (2.16)$$

By the definitions of $\mathcal{D}(\cdot)$ (Definition 2.5) and $d(\cdot)$ (Proposition 2.4), we have

$$\begin{aligned} \mathcal{D}(\boldsymbol{\lambda}^m, \nabla L(\boldsymbol{\lambda}^m), \mu_{\boldsymbol{D}^\star}) &= \sum_{\omega \in \Omega} d(\lambda_\omega^m, \nabla_\omega^m, \mu_{\boldsymbol{D}^\star}) \\ &= \sum_{\omega \in W^m} d(\lambda_\omega^m, \nabla_\omega^m, \mu_{\boldsymbol{D}^\star}) + \sum_{\omega \in \Omega \setminus W^m} d(\lambda_\omega^m, \nabla_\omega^m, \mu_{\boldsymbol{D}^\star}) \end{aligned} \quad (2.17)$$

Inexact Case: By the definition of $\boldsymbol{\lambda}^{W^m}$, we know that $\lambda_\omega^m = 0$ for $\omega \notin W^m$. Then it follows from Proposition 2.4 that

$$\mathcal{D}(\boldsymbol{\lambda}^m, \nabla L(\boldsymbol{\lambda}^m), \mu_{\boldsymbol{D}^\star}) = \sum_{\omega \in W^m} d(\lambda_\omega^m, \nabla_\omega^m, \mu_{\boldsymbol{D}^\star}) + \sum_{\omega \in \Omega \backslash W^m} (\tilde{\nabla}_\omega^m)^2 \tag{2.18}$$

Exact Case: By the primal feasibility, dual feasibility and complementary slackness of $\boldsymbol{\lambda}_{W^m}^\star$, we have $\lambda_\omega^m \nabla_\omega^m = 0$ with $\lambda_\omega^m, \nabla_\omega^m \leq 0$ for $\omega \in W^m$. Combining this with $\lambda_\omega^m = 0$ for $\omega \notin W^m$, by Proposition 2.4, we have

$$\mathcal{D}(\boldsymbol{\lambda}^m, \nabla L(\boldsymbol{\lambda}^m), \mu_{\boldsymbol{D}^\star}) = \sum_{\omega \in \Omega \backslash W^m} (\tilde{\nabla}_\omega^m)^2. \tag{2.19}$$

The conclusions from (2.18) and (2.19) lead to (2.15).

Step 2: In the second step, we will show that

$$\begin{aligned} & L(\boldsymbol{\lambda}^{m+1}) - L(\boldsymbol{\lambda}^m) \\ \leq & \begin{cases} -\dfrac{1}{2\sigma} \displaystyle\sum_{\omega \in W^{m+1} \backslash W^m} (\tilde{\nabla}_\omega^m)^2 & \text{(exact case)} \\ -\dfrac{1}{2\sigma} \left[\displaystyle\sum_{\omega \in W^m} d(\lambda_\omega^m, \nabla_\omega^m, \sigma) + \sum_{\omega \in W^{m+1} \backslash W^m} (\tilde{\nabla}_\omega^m)^2 \right] & \text{(inexact case)} \end{cases} \end{aligned} \tag{2.20}$$

Exact Case: Let $\Lambda^{m+1} = \{\boldsymbol{\lambda} \in \mathbb{R}^{n(n-1)} : \lambda_\omega = 0, \forall \omega \in \Omega \backslash W^{m+1}\}$. Recall that $\boldsymbol{\lambda}^{m+1}$ minimizes $L(\boldsymbol{\lambda})$ over Λ^{m+1}. Therefore, we have the following:

$$\begin{aligned} L(\boldsymbol{\lambda}^{m+1}) &= \min_{\boldsymbol{\lambda} \in \Lambda^{m+1}} L(\boldsymbol{\lambda}) \\ &\leq \min_{\boldsymbol{\lambda} \in \Lambda^{m+1}} \left\{ L(\boldsymbol{\lambda}^m) + \langle \nabla L(\boldsymbol{\lambda}^m), \boldsymbol{\lambda} - \boldsymbol{\lambda}^m \rangle + \frac{\sigma}{2} \|\boldsymbol{\lambda} - \boldsymbol{\lambda}^m\|^2 \right\} \\ &= L(\boldsymbol{\lambda}^m) - \frac{1}{2\sigma} \sum_{\omega \in W^m} d(\lambda_\omega^m, \nabla_\omega^m, \sigma) - \frac{1}{2\sigma} \sum_{\omega \in W^{m+1} \backslash W^m} d(\lambda_\omega^m, \nabla_\omega^m, \sigma) \\ &= L(\boldsymbol{\lambda}^m) - \frac{1}{2\sigma} \sum_{\omega \in W^{m+1} \backslash W^m} (\tilde{\nabla}_\omega^m)^2 \end{aligned} \tag{2.21}$$

where the first inequality uses σ-smoothness of L; the last line follows from an argument similar to (2.19) where we use Proposition 2.4 and complementary slackness.

Inexact Case: Here we take one (or more) projected gradient step(s) to partially minimize the reduced dual. Let $\boldsymbol{\lambda}^{m+\frac{1}{2}}$ be obtained after one projected gradient descent step from $\boldsymbol{\lambda}^m$. Then we have

$$\begin{aligned} L(\boldsymbol{\lambda}^{m+1}) &\leq L(\boldsymbol{\lambda}^{m+\frac{1}{2}}) \\ &\leq L(\boldsymbol{\lambda}^m) + \langle \nabla L(\boldsymbol{\lambda}^m), \boldsymbol{\lambda}^{m+\frac{1}{2}} - \boldsymbol{\lambda}^m \rangle + \frac{\sigma}{2} \|\boldsymbol{\lambda}^{m+\frac{1}{2}} - \boldsymbol{\lambda}^m\|^2 \\ &= L(\boldsymbol{\lambda}^m) - \frac{1}{2\sigma} \sum_{\omega \in W^{m+1}} d(\lambda_\omega^m, \nabla_\omega^m, \sigma) \\ &= L(\boldsymbol{\lambda}^m) - \frac{1}{2\sigma} \sum_{\omega \in W^m} d(\lambda_\omega^m, \nabla_\omega^m, \sigma) - \frac{1}{2\sigma} \sum_{\omega \in W^{m+1} \setminus W^m} (\tilde{\nabla}_\omega^m)^2, \end{aligned} \tag{2.22}$$

where the first inequality follows from the descent property of projected gradient descent; the second inequality uses σ-smoothness of L; and the last line follows from Proposition 2.4.

Finally, the result in (2.20) follows from (2.21) and (2.22)

Step 3: For the third step, we will show that the following holds

$$\sigma\alpha\mathbb{E}_{\eta_{m+1}}[L(\boldsymbol{\lambda}^{m+1}) - L(\boldsymbol{\lambda}^m)] \leq -\mu_{\boldsymbol{D}^\star}[L(\boldsymbol{\lambda}^m) - L(\boldsymbol{\lambda}^\star)]. \tag{2.23}$$

Exact Case: For this case, we have the following chain of inequalities:

$$\begin{aligned} -2\sigma\alpha\mathbb{E}_{\eta_{m+1}}[L(\boldsymbol{\lambda}^{m+1}) - L(\boldsymbol{\lambda}^m)] &\geq \alpha\mathbb{E}_{\eta_{m+1}} \sum_{\omega \in W^{m+1} \setminus W^m} (\tilde{\nabla}_\omega^m)^2 \\ &\geq \sum_{\omega \in \Omega \setminus W^m} (\tilde{\nabla}_\omega^m)^2 \\ &\geq 2\mu_{\boldsymbol{D}^\star}[L(\boldsymbol{\lambda}^m) - L(\boldsymbol{\lambda}^\star)], \end{aligned} \tag{2.24}$$

where the first inequality uses (2.20); the second inequality is the assumption (2.14); and the last line uses (2.15).

Inexact Case: Since $\lambda_\omega^m \leq 0$, we know $d(\lambda_\omega^m, \nabla_\omega^m, \sigma) \geq 0$ by Proposition 2.4. Using the fact that $\alpha \geq 1$, we have the following:

$$\begin{aligned}
-2\sigma\alpha\mathbb{E}_{\eta_{m+1}}[L(\boldsymbol{\lambda}^{m+1}) - L(\boldsymbol{\lambda}^m)] &\geq \alpha \sum_{\omega \in W^m} d(\lambda_\omega^m, \nabla_\omega^m, \sigma) + \alpha\mathbb{E}_{\eta_{m+1}} \sum_{\omega \in W^{m+1}\backslash W^m} (\tilde{\nabla}_\omega^m)^2 \\
&\geq \sum_{\omega \in W^m} d(\lambda_\omega^m, \nabla_\omega^m, \sigma) + \sum_{\omega \in \Omega\backslash W^m} (\tilde{\nabla}_\omega^m)^2 \\
&\geq 2\mu_{\boldsymbol{D}^\star}[L(\boldsymbol{\lambda}^m) - L(\boldsymbol{\lambda}^\star)], \qquad (2.25)
\end{aligned}$$

where the first inequality uses (2.20); the second inequality uses assumption (2.14), $\alpha \geq 1$ and $d(\lambda_\omega^m, \nabla_\omega^m, \sigma) \geq 0$; and the last line uses (2.15).

The statement in (2.23) follows from (2.24) and (2.25).

Finally, we complete the proof by using (2.23) and observing that:

$$\begin{aligned}
\mathbb{E}_{\eta_{m+1}}[L(\boldsymbol{\lambda}^{m+1}) - L^\star] =& \mathbb{E}_{\eta_{m+1}}[L(\boldsymbol{\lambda}^{m+1}) - L(\boldsymbol{\lambda}^m)] + L(\boldsymbol{\lambda}^m) - L^\star \\
&\leq \left(1 - \frac{\mu_{\boldsymbol{D}^\star}}{\alpha\sigma}\right)[L(\boldsymbol{\lambda}^m) - L^\star].
\end{aligned}$$

□

2.A.3 Proof of Theorem 2.1

The proof of Theorem 2.1 uses Lemmas 2.1 and 2.4.

Proof of Theorem 2.1. Recall that λ_ω^m is the ω-th component of $\boldsymbol{\lambda}^m$, and we denote by $\nabla_\omega^m = \nabla_{\lambda_\omega} L(\boldsymbol{\lambda}^m)$, and $\tilde{\nabla}_\omega^m = \max\{\nabla_\omega^m, 0\}$. Now let $\{\Omega_i\}$ be one of the partitions $\{\Omega_{i\cdot}\}$ or $\{\Omega_{\cdot i}\}$, and W_i^m be the corresponding partition for W^m. Let $\Delta = W^{m+1}\backslash W^m$. Using this notation, the condition of Lemma 2.4 reduces to

$$\alpha\mathbb{E}_{\eta_{m+1}}[\sum_{\omega \in \Delta} (\tilde{\nabla}_\omega^m)^2] \geq \sum_{\omega \in \Omega\backslash W^m} (\tilde{\nabla}_\omega^m)^2. \qquad (2.26)$$

We organize the proof into four steps: (1) we construct a random vector $\boldsymbol{g} \in \mathbb{R}^{n(n-1)}$ from $\boldsymbol{\nabla}^m = \nabla L(\boldsymbol{\lambda}^m)$; (2) we then show $\sum_{\omega \in \Omega\backslash W^m} (\tilde{\nabla}_\omega^m)^2$ equals $\|\boldsymbol{g}\|^2$; (3) we

relate $\mathbb{E}_{\eta_{m+1}}[\sum_{\omega\in\Delta}(\tilde{\nabla}_\omega^m)^2]$ to $\|\boldsymbol{g}\|_{\{\ell\}}^2$; and (4) finally, we apply Lemmas 2.1 and 2.4 to complete the proof.

Step 1: Define each entry g_ω of $\boldsymbol{g}$ as follows:

$$g_\omega = \begin{cases} \tilde{\nabla}_\omega^m & \text{if } \omega \in \Omega\backslash W^m \\ 0 & \text{if } \omega \in W^m \end{cases}. \tag{2.27}$$

Notice $\boldsymbol{g}$ is a random vector depending upon random sources from iterations $1, \ldots, m$.

Step 2: By the definition of $\boldsymbol{g}$, we have $\sum_{\omega\in\Omega\backslash W^m}(\tilde{\nabla}_\omega^m)^2 = \|\boldsymbol{g}\|_2^2$.

Step 3: Recall that Δ' denotes the set of selected pairs as per a rule, and we consider a subset $\Delta = \{\omega \in \Delta' : \tilde{\nabla}_\omega^m = \nabla_\omega^m > 0\}$—that is, $\tilde{\nabla}_\omega^m = 0$ for any $\omega \in \Delta'\backslash\Delta$. Thus,

$$\mathbb{E}_{\eta_{m+1}}\left[\sum_{\omega\in\Delta}(\tilde{\nabla}_\omega^m)^2\right] = \mathbb{E}_{\eta_{m+1}}\left[\sum_{\omega\in\Delta'}(\tilde{\nabla}_\omega^m)^2\right].$$

Note that $\Delta' = \delta_\ell(\tilde{\boldsymbol{\nabla}}_{\Omega\backslash W^m}^m, \{\Omega_i \setminus W_i^m\})$, and let $\Delta'' = \delta_\ell(\boldsymbol{g}, \{\Omega_i\})$. Notice that $\boldsymbol{g}$ has more zero coordinates compared to $\tilde{\boldsymbol{\nabla}}^m$ (see (2.27)). Thus, we have

$$\mathbb{E}_{\eta_{m+1}}\left[\sum_{\omega\in\Delta}(\tilde{\nabla}_\omega^m)^2\right] = \mathbb{E}_{\eta_{m+1}}\left[\sum_{\omega\in\Delta'}(\tilde{\nabla}_\omega^m)^2\right] \geq \mathbb{E}_{\eta_{m+1}}\left[\sum_{\omega\in\Delta''}(g_\omega)^2\right] = \|\boldsymbol{g}\|_{\{\ell\}}^2,$$

where the last equality follows from Definition 2.4.

Step 4: From Step 2, Step 3 and Lemma 2.1, we arrive at (2.26). Therefore, it follows from Lemma 2.4 that

$$\mathbb{E}_{\eta_{m+1}}[L(\boldsymbol{\lambda}^m) - L^\star] \leq \left(1 - \frac{\mu_{\boldsymbol{D}^\star}}{\alpha_{\{\ell\}}\sigma}\right)[L(\boldsymbol{\lambda}^m) - L^\star]$$

and (using tower property of expectation) we arrive at the conclusion of the theorem:

$$\mathbb{E}[L(\boldsymbol{\lambda}^m) - L^\star] \leq \left(1 - \frac{\mu_{\boldsymbol{D}^\star}}{\alpha_{\{\ell\}}\sigma}\right)^m [L(\boldsymbol{\lambda}^0) - L^\star].$$

□

Remark 2.4. *Both accelerated gradient methods (APG) and L-BFGS can be used to solve the reduced problems to optimality—so, the proof for the exact subproblem optimization (in Theorem 2.1) for these cases will be the same as that for PGD, The proof for the inexact subproblem optimization for PGD uses only the sufficient decrease condition of the first PGD update and the fact that PGD is a descent algorithm. Since APG with adaptive restart (function scheme) [179] and L-BFGS [152] are descent algorithms, the theory presented above, goes through as long as the progress made by APG or L-BFGS is at least as large as the progress made by the first step of APG. To this end, for L-BFGS, when setting the initial inverse Hessian approximation as the identity matrix, i.e.* $\boldsymbol{H}_0 = \boldsymbol{I}$*, the first step of L-BFGS is exactly PGD update, so the theory works for L-BFGS. For APG with restarts, if we perform a PGD update preceding the APG updates, then the theory will go through as well.*

2.B Additional Technical Details

2.B.1 Examples of unavoidable factor $\alpha_{\{\ell\}}$

For simplicity, we consider an unconstrained problem with objective $f(x) = x^\top H x$, where $H = \mathrm{diag}(\sigma, 1, 1, \ldots, 1) \in \mathbb{R}^{p\times p}$ with $\sigma > 1$. Then, f is 1-strongly convex and σ-smooth, and thus the step size will be $1/\sigma$.

Gradient Descent: Starting from any $x^k = (x_i^k : i \in [p])$, the gradient step yields

$$x^{k+1} = (0, \gamma x_2^k, \ldots, \gamma x_m^k),$$

where $\gamma = 1 - \frac{1}{\sigma}$. Therefore, for $k \geq 1$, $x_1^{k+1} = x_1^k = 0$, and $f(x^{k+1}) = \gamma^2 f(x^k)$. Since the minimum objective value $f^* = 0$, we have

$$f(x^{k+1}) - f^* = (1 - (1 - \gamma^2))(f(x^k) - f^*) \tag{2.28}$$

Our algorithm (inexact optimization): We now consider the inexact active set algorithm (denoted by ASGD). Suppose, we start with initial active set $\mathcal{W}^0 = \{1\}$ and

$x^0 = (1, 1, \ldots, 1)$. We take a gradient step over $\mathcal{W}^0$ to obtain $x^1 = (0, 1, \ldots, 1)$. Starting with $\mathcal{W}^{k-1}$ and x^k, we randomly select i_k from $[p] \backslash \mathcal{W}^{k-1}$, and augment $\mathcal{W}^{k-1}$ with i_k, i.e. $\mathcal{W}^k = \mathcal{W}^{k-1} \cup \{i_k\}$. Then, we take a gradient step from x^k over $\mathcal{W}^k$ to obtain x^{k+1}, it is easy to see that

$$x_i^{k+1} = \gamma x_i^k, \forall i \in \mathcal{W}^k, \quad \text{and} \quad x_i^{k+1} = x_i^k, \forall i \notin \mathcal{W}^k.$$

Moreover, by induction, we have that $x_1^{k+1} = 0$, x^{k+1} contains $(p-k-1)$ coordinates of 1, and k coordinates that are $\gamma, \gamma^2, \ldots, \gamma^k$. Therefore,

$$f(x^k) = (p-k-1) + \sum_{j=1}^{k} \gamma^{2k} = (p-k-1) + \frac{\gamma^2 - \gamma^{2k+2}}{1-\gamma^2}.$$

Since $f^* = 0$, we have for $k \geq 1$

$$\frac{f(x^k) - f(x^{k+1})}{f(x^k)} = \frac{1-\gamma^{2k+2}}{p-k-1+\frac{\gamma^2-\gamma^{2k+2}}{1-\gamma^2}} \geq \frac{1-\gamma^2}{p-1}.$$

The above inequality is tight, and it indicates

$$f(x^{k+1}) - f^* \leq \left(1 - \frac{1-\gamma^2}{p-1}\right)(f(x^k) - f^*) \tag{2.29}$$

Compared to the rate of gradient descent in (2.28), the above rate (2.29) has an additional $O(\frac{1}{p})$ factor. This is similar to having the factor $\alpha_{\{\ell\}}$ for the augmentation rules in Theorem 2.1.

2.C Additional Experiment Details

2.C.1 Real dataset details

RD1, RD2: These are taken from `https://archive.ics.uci.edu/ml/datasets/Gas+Turbine+CO+and+NOx+Emission+Data+Set`, and were studied in [130].

The dataset has 36,733 samples. RD1 has CO as response with features: AP, AFDP,

GTEP, CDP. RD2 has NOx as response with features: AT, AP, AH, AFDP. We apply the log-transform on the responses. Each training set has $n = 10{,}000$ and $d = 4$, with the remaining set aside for testing.

RD3, RD4: These two datasets are taken from `https://archive.ics.uci.edu/ml/datasets/Beijing+Multi-Site+Air-Quality+Data`.

They have been studied in [229]. This dataset has approximately 420,768 samples; we select SO2, NO2, CO, O3 as features. We apply the log-transform on all features and both responses PM2.5 and PM10. Each training dataset has $n = 10{,}000$ and $d = 4$ (remaining samples used for testing).

RD5: This dataset is taken from `https://archive.ics.uci.edu/ml/datasets/Beijing+PM2.5+Data`; and has been studied in [148]. We use PM2.5 as response with features: DEWP, TEMP, PRES, Iws. We use a log-transform of all features and response; and consider a training set with $n = 10{,}000$ and $d = 4$.

RD6: This dataset is taken from `https://archive.ics.uci.edu/ml/datasets/combined+cycle+power+plant`; and has been studied in [131, 208]. We consider a training set with $n = 5{,}000$ and $d = 4$.

RD7: This dataset taken from [164], is available at `http://ampd.epa.gov/ampd/`, and has been recently used in [158, 31]. We apply the log-transformation on all features and the response, and consider a training set with $n = 30{,}000$ and $d = 4$.

RD8: This is taken from [186] and is the `ex1029` dataset available from `R` package `Sleuth2`—see also [112, 14]. We first winsorize the data by excluding the samples that have response with score ≥ 2; and consider a training set with $n = 15{,}000$ and $d = 4$. Following [14], we apply the transformation $x \mapsto 1.2^x$ to the education variable.

2.C.2 Algorithm Parameters

The tolerance for violations of constraints is set as 10^{-4} (10^{-8} for the second stage). For inexact optimization, the tolerance for the relative objective change is 10^{-6} and the maximum number of PGD iterations is taken to be 5. For exact optimization, the violation tolerance is 10^{-7}, the maximum number of PGD iterations for the sub-problems is 3,000 and the minimum number of PGD iterations for the sub-problems

is 5.

In Rule 2, we take $K = n$; in Rule 4, we take $M = 4n$ and $K = n$. In Rule 1, we take $P = 1$; in Rule 5, we take $G = n/4$, and $P = 4$.

In the second stage, we apply the occasional rule 1/5 when: (i) the number of constraints added by random rules 2/4 is less than $0.005n$ for consecutive 5 iterations; or (ii) the number of PGD iterations in the subproblems is the minimum number 5 for consecutive 5 iterations.

Scripts used to run the experiments containing all algorithm parameters can be found in our github repository.

Chapter 3

A New Computational Framework for Log-concave Density Estimation

This chapter is based on [53]. It is a joint work with Rahul Mazumder and Richard J. Samworth.

3.1 Introduction

In Statistics, the field of nonparametric inference under shape constraints dates back at least to [99], who studied the nonparametric maximum likelihood estimator of a decreasing density on the non-negative half line. But it is really over the last decade or so that researchers have begun to realize its full potential for addressing key contemporary data challenges such as (multivariate) density estimation and regression. The initial allure is the flexibility of a nonparametric model, combined with estimation methods that can often avoid the need for tuning parameter selection, which can often be troublesome for other nonparametric techniques such as those based on smoothing. Intensive research efforts over recent years have revealed further great attractions: for instance, these procedures frequently attain optimal rates of convergence over relevant function classes. Moreover, it is now known that shape-constrained procedures can possess intriguing adaptation properties, in the sense that they can estimate particular subclasses of functions at faster rates, even (nearly) as well as the best one

could do if one were told in advance that the function belonged to this subclass.

Typically, however, the implementation of shape-constrained estimation techniques requires the solution of an optimization problem, and, despite some progress, there are several cases where computation remains a bottleneck and hampers the adoption of these methods by practitioners. In this chapter, we focus on the problem of log-concave density estimation, which has become arguably the central challenge in the field because the class of log-concave densities enjoys stability properties under marginalization, conditioning, convolution and linear transformations that make it a very natural infinite-dimensional generalization of the class of Gaussian densities [195].

The univariate log-concave density estimation problem was first studied in [214], and fast algorithms for the computation of the log-concave maximum likelihood estimator (MLE) in one dimension are now available through the `R` packages `logcondens` [77] and `cnmlcd` [153]. [61] introduced and studied the multivariate log-concave maximum likelihood estimator, but their algorithm, which is described below and implemented in the `R` package `LogConcDEAD` [59], is slow; for instance, [61] report a running time of 50 seconds for computing the bivariate log-concave MLE with 500 observations, and 224 minutes for computing the log-concave MLE in four dimensions with 2,000 observations. An alternative, interior point method for a suitable approximation was proposed by [137]. Recent progress on theoretical aspects of the computational problem in the computer science community includes [10], who proved that there exists a polynomial time algorithm for computing the log-concave maximum likelihood estimator. We are unaware of any attempt to implement this algorithm. [188] compute an approximation to the log-concave MLE by considering $-\log p$ as a piecewise affine maximum function, using the log-sum-exp operator to approximate the non-smooth operator, a Riemann sum to compute the integral and its gradient, and obtain a solution via L-BFGS. This reformulation means that the problem is no longer convex.

To describe the problem more formally, let $\mathcal{C}_d$ denote the class of proper, convex lower-semicontinuous functions $\varphi : \mathbb{R}^d \to (-\infty, \infty]$ that are coercive in the sense that

$\varphi(\boldsymbol{x}) \to \infty$ as $\|\boldsymbol{x}\| \to \infty$. The class of upper semi-continuous log-concave densities on $\mathbb{R}^d$ is denoted as

$$\mathcal{P}_d := \Big\{ p : \mathbb{R}^d \to [0, \infty) : p = e^{-\varphi} \text{ for some } \varphi \in \mathcal{C}_d, \int_{\mathbb{R}^d} p = 1 \Big\}.$$

Given $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathbb{R}^d$, [61, Theorem 1] proved that whenever the convex hull C_n of $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ is d-dimensional, there exists a unique

$$\hat{p}_n \in \operatorname*{argmax}_{p \in \mathcal{P}_d} \frac{1}{n} \sum_{i=1}^{n} \log p(\boldsymbol{x}_i). \tag{3.1}$$

If $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ are regarded as realizations of independent and identically distributed random vectors on $\mathbb{R}^d$, then the objective function in (3.1) is a scaled version of the log-likelihood function, so $\hat{p}_n$ is called the *log-concave MLE*. The existence and uniqueness of this estimator is not obvious, because the infinite-dimensional class $\mathcal{P}_d$ is non-convex, and even the class of negative log-densities $\big\{\varphi \in \mathcal{C}_d : \int_{\mathbb{R}^d} e^{-\varphi} = 1\big\}$ is non-convex. In fact, the estimator belongs to a finite-dimensional subclass; more precisely, for a vector $\boldsymbol{\phi} = (\phi_1, \ldots, \phi_n) \in \mathbb{R}^n$, define $\mathrm{cef}[\boldsymbol{\phi}] \in \mathcal{C}_d$ to be the (pointwise) largest function with

$$\mathrm{cef}[\boldsymbol{\phi}](\boldsymbol{x}_i) \leq \phi_i$$

for $i = 1, \ldots, n$. [61] proved that $\hat{p}_n = e^{-\mathrm{cef}[\boldsymbol{\phi}^*]}$ for some $\boldsymbol{\phi}^* \in \mathbb{R}^n$, and refer to the function $-\mathrm{cef}[\boldsymbol{\phi}^*]$ as a 'tent function'; see the illustration in Figure 3-1. [61] further defined the non-smooth, convex objective function $f : \mathbb{R}^n \to \mathbb{R}$ by

$$f(\boldsymbol{\phi}) \equiv f(\phi_1, \ldots, \phi_n) := \frac{1}{n} \sum_{i=1}^{n} \phi_i + \int_{C_n} \exp\big\{-\mathrm{cef}[\boldsymbol{\phi}](x)\big\}\, dx, \tag{3.2}$$

and proved that $\boldsymbol{\phi}^* = \operatorname{argmin}_{\boldsymbol{\phi} \in \mathbb{R}^n} f(\boldsymbol{\phi})$.

The two main challenges in optimizing the objective function f in (3.2) are that the value and subgradient of the integral term are hard to evaluate, and that it is non-smooth, so vanilla subgradient methods lead to a slow rate of convergence. To address the first issue, [61] computed the exact integral and its subgradient using

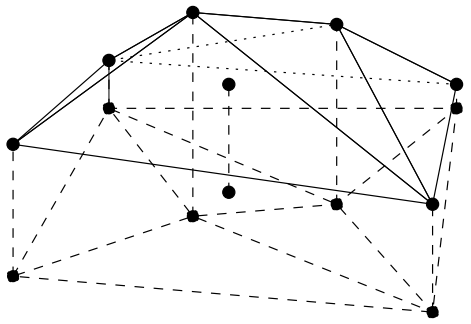

Figure 3-1: An illustration of a tent function, taken from [61].

the `qhull` algorithm [15] to obtain a triangulation of the convex hull of the data, evaluating the function value and subgradient over each simplex in the triangulation. However, in the worst case, the triangulation can have $O(n^{d/2})$ simplices [162]. The non-smoothness is handled via Shor's r-algorithm [203, Chapter 3], as implemented by [128].

In Section 3.2, we characterize the subdifferential of the objective function in terms of the solution of a linear program (LP), and show that the solution lies in a known, compact subset of $\mathbb{R}^n$. This understanding allows us to introduce our new computational framework for log-concave density estimation in Section 3.3, based on an accelerated version of a dual averaging approach [174]. This relies on smoothing the objective function, and encompasses two popular strategies, namely Nesterov smoothing [172] and randomized smoothing [143, 225, 75], as special cases. A further feature of our algorithm is the construction of approximations to gradients of our smoothed objective, and this in turn requires an approximation to the integral in (3.2). While a direct application of the theory of [75] would yield a rate of convergence for the objective function of order $n^{1/4}/T + 1/\sqrt{T}$ after T iterations, we show in Section 3.4 that by introducing finer approximations of both the integral and its gradient as the iteration number increases, we can obtain an improved rate of order $1/T$, up to logarithmic factors. Moreover, we translate the optimization error in the objective into a bound on the error in the log-density, which is uncommon in the literature in the absence of strong convexity. A further advantage of our approach is that we are able to extend it in Section 3.5 to the more general problem of quasi-concave density

estimation [137, 201], thereby providing a computationally tractable alternative to the discrete Hessian approach of [137]. Section 3.6 illustrates the practical benefits of our methodology in terms of improved computational timings on simulated data. Additional experimental details and applications on real data sets are provided in Appendix 3.A. Proofs of all main results can be found in Appendix 3.B, and background on the field of nonparametric inference under shape constraints can be found in Appendix 3.C.

Notation: We write $[n] := \{1, 2, \ldots, n\}$, let $\mathbf{1} \in \mathbb{R}^n$ denote the all-ones vector, and denote the cardinality of a set S by $|S|$. For a Borel measurable set $C \subseteq \mathbb{R}^d$, we use $\mathrm{vol}(C)$ to denote its volume (i.e. d-dimensional Lebesgue measure). We write $\|\cdot\|$ for the Euclidean norm of a vector. For $\mu > 0$, a convex function $f : \mathbb{R}^n \to \mathbb{R}$ is said to be μ-strongly convex if $\boldsymbol{\phi} \mapsto f(\boldsymbol{\phi}) - \frac{\mu}{2}\|\boldsymbol{\phi}\|^2$ is convex. The notation $\partial f(\boldsymbol{\phi})$ denotes the subdifferential (set of subgradients) of f at $\boldsymbol{\phi}$. Given a real-valued sequence (a_n) and a positive sequence (b_n), we write $a_n = \tilde{O}(b_n)$ if there exist $C, \gamma > 0$ such that $a_n \leq C b_n \log^{\gamma}(1 + n)$ for all $n \in \mathbb{N}$.

3.2 Understanding the structure of the optimization problem

Throughout this chapter, we assume that $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathbb{R}^d$ are distinct and that their convex hull $C_n := \mathrm{conv}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$ has nonempty interior, so that $n \geq d + 1$ and $\Delta := \mathrm{vol}(C_n) > 0$. This latter assumption ensures the existence and uniqueness of a minimizer of the objective function in (3.2) [78, Theorem 2.2]. Recall that we define the *lower convex envelope* function [193] $\mathrm{cef} : \mathbb{R}^n \to \mathcal{C}_d$ by

$$\mathrm{cef}[\boldsymbol{\phi}](\boldsymbol{x}) \equiv \mathrm{cef}[(\phi_1, \ldots, \phi_n)](\boldsymbol{x}) := \sup\big\{g(\boldsymbol{x}) : g \in \mathcal{C}_d, g(\boldsymbol{x}_i) \leq \phi_i \ \forall i \in [n]\big\}. \quad (3.3)$$

As mentioned in the introduction, in computing the MLE, we seek

$$\boldsymbol{\phi}^* := \operatorname*{argmin}_{\boldsymbol{\phi}\in\mathbb{R}^n} f(\boldsymbol{\phi}), \tag{3.4}$$

where

$$f(\boldsymbol{\phi}) := \frac{1}{n}\mathbf{1}^\top\boldsymbol{\phi} + \int_{C_n} \exp\{-\mathrm{cef}[\boldsymbol{\phi}](\boldsymbol{x})\}\, d\boldsymbol{x} =: \frac{1}{n}\mathbf{1}^\top\boldsymbol{\phi} + I(\boldsymbol{\phi}). \tag{3.5}$$

Note that (3.4) can be viewed as a stochastic optimization problem by writing

$$f(\boldsymbol{\phi}) = \mathbb{E}F(\boldsymbol{\phi}, \boldsymbol{\xi}), \tag{3.6}$$

where $\boldsymbol{\xi}$ is uniformly distributed on C_n and where, for $\boldsymbol{x} \in C_n$,

$$F(\boldsymbol{\phi}, \boldsymbol{x}) := \frac{1}{n}\mathbf{1}^\top\boldsymbol{\phi} + \Delta e^{-\mathrm{cef}[\boldsymbol{\phi}](\boldsymbol{x})}. \tag{3.7}$$

Let $\boldsymbol{X} := [\boldsymbol{x}_1 \cdots \boldsymbol{x}_n]^\top \in \mathbb{R}^{n\times d}$, and for $\boldsymbol{x} \in \mathbb{R}^d$, let $E(\boldsymbol{x}) := \{\boldsymbol{\alpha} \in \mathbb{R}^n : \boldsymbol{X}^\top\boldsymbol{\alpha} = \boldsymbol{x}, \mathbf{1}_n^\top\boldsymbol{\alpha} = 1, \boldsymbol{\alpha} \geq 0\}$ denote the set of all weight vectors for which $\boldsymbol{x}$ can be written as a weighted convex combination of $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$. Thus $E(\boldsymbol{x})$ is a compact, convex subset of $\mathbb{R}^n$. The cef function is given by a linear program (LP) [137, 10]:

$$\mathrm{cef}[\boldsymbol{\phi}](\boldsymbol{x}) = \inf_{\boldsymbol{\alpha}\in E(\boldsymbol{x})} \boldsymbol{\alpha}^\top\boldsymbol{\phi}. \tag{Q_0}$$

If $\boldsymbol{x} \notin C_n$, then $E(\boldsymbol{x}) = \emptyset$, and, with the standard convention that $\inf\emptyset := \infty$, we see that (Q_0) agrees with (3.3). From the LP formulation, it follows that $\boldsymbol{\phi} \mapsto \mathrm{cef}[\boldsymbol{\phi}](\boldsymbol{x})$ is concave, for every $\boldsymbol{x} \in \mathbb{R}^d$.

Given a pair $\boldsymbol{\phi} \in \mathbb{R}^n$ and $\boldsymbol{x} \in C_n$, an optimal solution to (Q_0) may not be unique, in which case the map $\boldsymbol{\phi} \mapsto \mathrm{cef}[\boldsymbol{\phi}](\boldsymbol{x})$ is not differentiable [27, Proposition B.25(b)]. Noting that the infimum in (Q_0) is attained whenever $\boldsymbol{x} \in C_n$, let

$$\begin{aligned} A[\boldsymbol{\phi}](\boldsymbol{x}) &:= \mathrm{conv}\big(\{\boldsymbol{\alpha} \in E(\boldsymbol{x}) : \boldsymbol{\alpha}^\top\boldsymbol{\phi} = \mathrm{cef}[\boldsymbol{\phi}](\boldsymbol{x})\}\big) \\ &= \{\boldsymbol{\alpha} \in E(\boldsymbol{x}) : \boldsymbol{\alpha}^\top\boldsymbol{\phi} = \mathrm{cef}[\boldsymbol{\phi}](\boldsymbol{x})\}. \end{aligned}$$

Danskin's theorem [27, Proposition B.25(b)] applied to $-\mathrm{cef}[\boldsymbol{\phi}](\boldsymbol{x})$ then yields that for each $\boldsymbol{x} \in C_n$, the subdifferential of $F(\boldsymbol{\phi}, \boldsymbol{x})$ with respect to $\boldsymbol{\phi}$ is given by

$$\partial F(\boldsymbol{\phi}, \boldsymbol{x}) := \Big\{ \frac{1}{n}\mathbf{1} - \Delta e^{-\mathrm{cef}[\boldsymbol{\phi}](\boldsymbol{x})}\boldsymbol{\alpha} : \boldsymbol{\alpha} \in A[\boldsymbol{\phi}](\boldsymbol{x}) \Big\}. \tag{3.8}$$

Since both f and $F(\cdot, \boldsymbol{x})$ are finite convex functions on $\mathbb{R}^n$ (for each fixed $\boldsymbol{x} \in C_n$ in the latter case), by [56, Proposition 2.3.6(b) and Theorem 2.7.2], the subdifferential of f at $\boldsymbol{\phi} \in \mathbb{R}^n$ is given by

$$\partial f(\boldsymbol{\phi}) := \big\{ \mathbb{E}\boldsymbol{G}(\boldsymbol{\phi}, \boldsymbol{\xi}) : \boldsymbol{G}(\boldsymbol{\phi}, \boldsymbol{x}) \in \partial F(\boldsymbol{\phi}, \boldsymbol{x}) \text{ for each } \boldsymbol{x} \in C_n \big\}. \tag{3.9}$$

Observe that given any $\boldsymbol{\phi} \in \mathbb{R}^n$, the function $\boldsymbol{x} \mapsto -\mathrm{cef}\big[\boldsymbol{\phi} + \log I(\boldsymbol{\phi})\mathbf{1}\big](\boldsymbol{x})$ (where $I(\boldsymbol{\phi})$ is the integral defined in (3.5)) is a log-density. It is also convenient to let $\bar{\boldsymbol{\phi}} \in \mathbb{R}^n$ be such that $\exp\{-\mathrm{cef}[\bar{\boldsymbol{\phi}}]\}$ is the uniform density on C_n, so that $f(\bar{\boldsymbol{\phi}}) = \log\Delta + 1$. Proposition 3.1 below (an extension of [10, Lemma 2]) provides uniform upper and lower bounds on this log-density, whenever the objective function f evaluated at $\boldsymbol{\phi}$ is at least as good as that at $\bar{\boldsymbol{\phi}}$. In more statistical language, these bounds hold whenever the log-likelihood of the density $\exp\big\{-\mathrm{cef}\big[\boldsymbol{\phi} + \log I(\boldsymbol{\phi})\mathbf{1}\big](\cdot)\big\}$ is at least as large as that of the uniform density on the convex hull of the data, so in particular, they must hold for the log-concave MLE (i.e. when $\boldsymbol{\phi} = \boldsymbol{\phi}^*$). Let $\phi^0 := (n-1) + d(n-1)\log\big(2n + 2nd\log(2nd)\big) + \log\Delta$ and $\phi_0 := -1 - d\log\big(2n + 2nd\log(2nd)\big) + \log\Delta$.

Proposition 3.1. *For any $\boldsymbol{\phi} \in \mathbb{R}^n$ such that $f(\boldsymbol{\phi}) \leq \log\Delta + 1$, we have $\phi_0 \leq \phi_i + \log I(\boldsymbol{\phi}) \leq \phi^0$ for all $i \in [n]$.*

The following corollary is an immediate consequence of Proposition 3.1.

Corollary 3.1. *Suppose that $\boldsymbol{\phi} \in \mathbb{R}^n$ satisfies $I(\boldsymbol{\phi}) = 1$ and $f(\boldsymbol{\phi}) \leq f(\bar{\boldsymbol{\phi}}) = \log\Delta + 1$. Then $\boldsymbol{\phi}^* \in \mathbb{R}^n$ defined in (3.4) satisfies*

$$\|\boldsymbol{\phi} - \boldsymbol{\phi}^*\| \leq \sqrt{n}(\phi^0 - \phi_0).$$

Corollary 3.1 gives a sense in which any $\boldsymbol{\phi} \in \mathbb{R}^n$ for which the objective function is

'good' cannot be too far from the optimizer $\boldsymbol{\phi}^*$; here, 'good' means that the objective should be no larger than that of the uniform density on the convex hull of the data. Moreover, an upper bound on the integral $I(\boldsymbol{\phi})$ provides an upper bound on the norm of any subgradient $\boldsymbol{g}(\boldsymbol{\phi})$ of f at $\boldsymbol{\phi}$.

Proposition 3.2. *Any subgradient $\boldsymbol{g}(\boldsymbol{\phi}) \in \mathbb{R}^n$ of f at $\boldsymbol{\phi} \in \mathbb{R}^n$ satisfies $\|\boldsymbol{g}(\boldsymbol{\phi})\|^2 \leq \max\{1/n + 1/4, I(\boldsymbol{\phi})^2\}$.*

3.3 Computing the log-concave MLE

As mentioned in the introduction, subgradient methods [203, 185] tend to be slow for minimizing the objective function f defined in (3.5) [61]. Our alternative approach involves the minimizing the representation of f given in (3.6) via smoothing techniques, which offer superior computational guarantees and practical performance in our numerical experiments.

3.3.1 Smoothing techniques

We present two smoothing techniques to find the minimizer $\boldsymbol{\phi}^* \in \mathbb{R}^n$ of the nonsmooth convex optimization problem (3.4). By Proposition 3.1, we have that $\boldsymbol{\phi}^* \in \boldsymbol{\Phi}$, where

$$\boldsymbol{\Phi} := \{\boldsymbol{\phi} = (\phi_1, \ldots, \phi_n) \in \mathbb{R}^n : \phi_0 \leq \phi_i \leq \phi^0 \text{ for } i \in [n]\}, \tag{3.10}$$

with $\phi_0, \phi^0 \in \mathbb{R}$. In what follows we present two smoothing techniques: one based on Nesterov smoothing [172] and the second on randomized smoothing [75].

3.3.1.1 Nesterov smoothing

Recall that the non-differentiability in f in (3.5) is due to the LP (Q_0) potentially having multiple optimal solutions. Therefore, following [172], we consider replacing this LP with the following quadratic program (QP):

$$q_u[\boldsymbol{\phi}](\boldsymbol{x}) := \inf_{\boldsymbol{\alpha} \in E(\boldsymbol{x})} \left(\boldsymbol{\alpha}^\top \boldsymbol{\phi} + \frac{u}{2} \|\boldsymbol{\alpha} - \boldsymbol{\alpha}_0\|^2 - \frac{u}{2} \right), \tag{Q_u}$$

where $\boldsymbol{\alpha}_0 := (1/n)\mathbf{1} \in \mathbb{R}^n$ is the center of $E(\boldsymbol{x})$, and where $u \geq 0$ is a regularization parameter that controls the extent of the quadratic regularization of the objective. With this definition, we have $q_0[\boldsymbol{\phi}](\boldsymbol{x}) = \mathrm{cef}[\boldsymbol{\phi}](\boldsymbol{x})$. For $u > 0$, due to the strong convexity of the function $\boldsymbol{\alpha} \mapsto \boldsymbol{\alpha}^\top \boldsymbol{\phi} + (u/2)\|\boldsymbol{\alpha} - \boldsymbol{\alpha}_0\|^2$ on the convex polytope $E(\boldsymbol{x})$, (Q_u) admits a unique solution that we denote by $\boldsymbol{\alpha}_u^*[\boldsymbol{\phi}](\boldsymbol{x})$. It follows again from Danskin's theorem that $\boldsymbol{\phi} \mapsto q_u[\boldsymbol{\phi}](\boldsymbol{x})$ is differentiable for such u, with gradient $\nabla_{\boldsymbol{\phi}} q_u[\boldsymbol{\phi}](\boldsymbol{x}) = \boldsymbol{\alpha}_u^*[\boldsymbol{\phi}](\boldsymbol{x})$.

Using $q_u[\boldsymbol{\phi}](\boldsymbol{x})$ instead of $q_0[\boldsymbol{\phi}](\boldsymbol{x})$ in (3.5), we obtain a smooth objective $\boldsymbol{\phi} \mapsto \tilde{f}_u(\boldsymbol{\phi})$, given by

$$\tilde{f}_u(\boldsymbol{\phi}) := \frac{1}{n}\mathbf{1}^\top \boldsymbol{\phi} + \int_{C_n} \exp\{-q_u[\boldsymbol{\phi}](\boldsymbol{x})\}\, d\boldsymbol{x} = \mathbb{E}\tilde{F}_u(\boldsymbol{\phi}, \boldsymbol{\xi}), \tag{3.11}$$

where $\tilde{F}_u(\boldsymbol{\phi}, \boldsymbol{x}) := (1/n)\mathbf{1}^\top \boldsymbol{\phi} + \Delta \exp\{-q_u[\boldsymbol{\phi}](\boldsymbol{x})\}$, and where $\boldsymbol{\xi}$ is again uniformly distributed on C_n. We may differentiate under the integral (e.g. [136, Theorem 6.28]) to see that the partial derivatives of $\tilde{f}_u$ with respect to each component of $\boldsymbol{\phi}$ exist, and moreover they are continuous (because $\boldsymbol{\phi} \mapsto \boldsymbol{\alpha}_u^*[\boldsymbol{\phi}](\boldsymbol{x})$ is continuous by Proposition 3.5), so $\nabla_{\boldsymbol{\phi}} \tilde{f}_u(\boldsymbol{\phi}) = \mathbb{E}[\tilde{\boldsymbol{G}}_u(\boldsymbol{\phi}, \boldsymbol{\xi})]$, where

$$\tilde{\boldsymbol{G}}_u(\boldsymbol{\phi}, \boldsymbol{x}) := \nabla_{\boldsymbol{\phi}} \tilde{F}_u(\boldsymbol{\phi}, \boldsymbol{x}) = \frac{1}{n}\mathbf{1} - \Delta e^{-q_u[\boldsymbol{\phi}](\boldsymbol{x})} \alpha_u^*[\boldsymbol{\phi}](\boldsymbol{x}). \tag{3.12}$$

Proposition 3.3 below presents some properties of the smooth objective $\tilde{f}_u$.

Proposition 3.3. *For any $\boldsymbol{\phi} \in \boldsymbol{\Phi}$, we have*

(a) $0 \leq \tilde{f}_u(\boldsymbol{\phi}) - \tilde{f}_{u'}(\boldsymbol{\phi}) \leq \frac{u-u'}{2} e^{u'/2} I(\boldsymbol{\phi})$ for $u' \in [0, u]$;

(b) For every $u \geq 0$, the function $\boldsymbol{\phi} \mapsto \tilde{f}_u(\boldsymbol{\phi})$ is convex and $\Delta e^{-\phi_0 + u/2}$-Lipschitz;

(c) For every $u \geq 0$, the function $\boldsymbol{\phi} \mapsto \tilde{f}_u(\boldsymbol{\phi})$ has $\Delta e^{-\phi_0+u/2}(1 + u^{-1})$-Lipschitz gradient;

(d) $\mathbb{E}\big(\|\tilde{\boldsymbol{G}}_u(\boldsymbol{\phi}, \boldsymbol{\xi}) - \nabla_{\boldsymbol{\phi}} \tilde{f}_u(\boldsymbol{\phi})\|^2\big) \leq (\Delta e^{-\phi_0+u/2})^2$ for every $u \geq 0$.

3.3.1.2 Randomized smoothing

Our second smoothing technique is randomized smoothing [143, 225, 75]: we take the expectation of a random perturbation of the argument of f. Specifically, for $u \geq 0$, let

$$\bar{f}_u(\boldsymbol{\phi}) := \mathbb{E} f(\boldsymbol{\phi} + u\boldsymbol{z}), \tag{3.13}$$

where $\boldsymbol{z}$ is uniformly distributed on the unit ℓ_2-ball in $\mathbb{R}^n$. Thus, similar to Nesterov smoothing, $\bar{f}_0 = f$, and the amount of smoothing increases with u. From a stochastic optimization viewpoint, we can write

$$\bar{f}_u(\boldsymbol{\phi}) = \mathbb{E} F(\boldsymbol{\phi} + u\boldsymbol{z}, \boldsymbol{\xi}) \quad \text{and} \quad \nabla_{\boldsymbol{\phi}} \bar{f}_u(\boldsymbol{\phi}) = \mathbb{E} \boldsymbol{G}(\boldsymbol{\phi} + u\boldsymbol{z}, \boldsymbol{\xi})$$

where $\boldsymbol{G}(\boldsymbol{\phi} + u\boldsymbol{v}, \boldsymbol{x}) \in \partial F(\boldsymbol{\phi} + u\boldsymbol{v}, \boldsymbol{x})$, and where the expectations are taken over independent random vectors $\boldsymbol{z}$, distributed uniformly on the unit Euclidean ball in $\mathbb{R}^n$, and $\boldsymbol{\xi}$, distributed uniformly on C_n. Here the gradient expression follows from, e.g., [143, Lemma 3.3(a)], [225, Lemma 7]; since $F(\boldsymbol{\phi} + u\boldsymbol{v}, \boldsymbol{x})$ is differentiable almost everywhere with respect to $\boldsymbol{\phi}$, the expression for $\bar{f}_u(\boldsymbol{\phi})$ does not depend on the choice of subgradient.

Proposition 3.4 below lists some properties of $\bar{f}_u$ and its gradient. It extends [225, Lemmas 7 and 8] by exploiting special properties of the objective function to sharpen the dependence of the bounds on n.

Proposition 3.4. *For any $u \geq 0$ and $\boldsymbol{\phi} \in \boldsymbol{\Phi}$, we have*

(a) $0 \leq \bar{f}_u(\boldsymbol{\phi}) - f(\boldsymbol{\phi}) \leq I(\boldsymbol{\phi}) u e^u \sqrt{\frac{2 \log n}{n+1}}$;

(b) $\bar{f}_{u'}(\boldsymbol{\phi}) \leq \bar{f}_u(\boldsymbol{\phi})$ *for any* $u' \in [0, u]$;

(c) $\boldsymbol{\phi} \mapsto \bar{f}_u(\boldsymbol{\phi})$ *is convex and* $\Delta e^{-\phi_0 + u}$*-Lipschitz;*

(d) $\boldsymbol{\phi} \mapsto \bar{f}_u(\boldsymbol{\phi})$ *has* $\Delta e^{-\phi_0 + u} n^{1/2}/u$*-Lipschitz gradient;*

(e) $\mathbb{E}\big(\big\|\boldsymbol{G}(\boldsymbol{\phi} + u\boldsymbol{z}, \boldsymbol{\xi}) - \nabla \bar{f}_u(\boldsymbol{\phi})\big\|^2\big) \leq (\Delta e^{-\phi_0 + u})^2$ *whenever* $\boldsymbol{G}(\boldsymbol{\phi} + u\boldsymbol{v}, \boldsymbol{x}) \in \partial F(\boldsymbol{\phi} + u\boldsymbol{v}, \boldsymbol{x})$ *for every* $\boldsymbol{v} \in \mathbb{R}^n$ *with* $\|\boldsymbol{v}\| \leq 1$ *and* $\boldsymbol{x} \in C_n$.

3.3.2 Stochastic first-order methods for smoothing sequences

Our proposed algorithm for computing the log-concave MLE is given in Algorithm 3.1. It relies on the choice of a smoothing sequence of f, which may be constructed using Nesterov or randomized smoothing, for instance. For a non-negative sequence $(u_t)_{t\in\mathbb{N}_0}$, this smoothing sequence is denoted by $(\ell_{u_t})_{t\in\mathbb{N}_0}$, where $\ell_{u_t} := \tilde{f}_{u_t}$ is given by (3.11) or $\ell_{u_t} := \bar{f}_{u_t}$ is given by (3.13). In Algorithm 3.1, $P_{\boldsymbol{\Phi}} : \mathbb{R}^n \to \boldsymbol{\Phi}$ denotes the projection operator onto the closed convex set $\boldsymbol{\Phi}$, which is essentially a threshold clipping operator. In fact, Algorithm 3.1 is a modification of an algorithm due to [75], and can be regarded as an accelerated version of the dual averaging scheme [174] applied to (ℓ_{u_t}).

Algorithm 3.1 Accelerated stochastic dual averaging on a smoothing sequence with increasing grids

Input: Smoothing sequence (ℓ_{u_t}) whose gradients have Lipschitz constants $(L_t)_{t\in\mathbb{N}_0}$; initialization $\boldsymbol{\phi}_0 \in \mathbb{R}^n$; learning rate sequence $(\eta_t)_{t\in\mathbb{N}}$ of positive real numbers; number of iterations $T \in \mathbb{N}$

1: $\boldsymbol{\phi}_0^{(x)} = \boldsymbol{\phi}_0^{(y)} = \boldsymbol{\phi}_0^{(z)} = \boldsymbol{\phi}_0$, $\boldsymbol{s}_t = \mathbf{0} \in \mathbb{R}^n$, $\theta_0 = 1$
2: **for** $t = 0, \dots, T-1$ **do**
3: Compute an approximation $\boldsymbol{g}_t$ of $\nabla_{\boldsymbol{\phi}} \ell_{u_t}\big(\boldsymbol{\phi}_t^{(y)}\big)$; see Section 3.3.2.1
4: $\boldsymbol{s}_{t+1} = \boldsymbol{s}_t + \boldsymbol{g}_t/\theta_t$
5: $\theta_{t+1} = \frac{2}{1+\sqrt{1+4/\theta_t^2}}$
6: $\boldsymbol{\phi}_{t+1}^{(z)} = P_{\boldsymbol{\Phi}}\big(\boldsymbol{\phi}_0 - \frac{\boldsymbol{s}_t}{L_{t+1}+\eta_{t+1}/\theta_{t+1}}\big)$
7: $\boldsymbol{\phi}_{t+1}^{(x)} = (1-\theta_t)\boldsymbol{\phi}_t^{(x)} + \theta_t \boldsymbol{\phi}_{t+1}^{(z)}$
8: $\boldsymbol{\phi}_{t+1}^{(y)} = (1-\theta_{t+1})\boldsymbol{\phi}_{t+1}^{(x)} + \theta_{t+1}\boldsymbol{\phi}_{t+1}^{(z)}$
9: **end for**
10: **return** $\boldsymbol{\phi}_{t+1}^{(x)}$

3.3.2.1 Approximating the gradient of the smoothing sequence

In Line 3 of Algorithm 3.1, we need to compute an approximation of the gradient $\nabla_{\boldsymbol{\phi}} \ell_u$, for a general $u \geq 0$. A key step in this process is to approximate the integral $I(\cdot)$, as well as a subgradient of I, at an arbitrary $\boldsymbol{\phi} \in \mathbb{R}^n$. [61] provide explicit formulae for these quantities, based on a triangulation of C_n, using tools from computational geometry. For practical purposes, [62] apply a Taylor expansion to approximate the

analytic expression. The `R` package `LogConcDEAD` [59] uses this method to evaluate the exact integral at each iteration, but since this is time-consuming, we will only use this method at the final stage of our proposed algorithm as a polishing step[1].

An alternative approach is to use numerical integration[2]. Among deterministic schemes, [188] observed empirically that the simple Riemann sum with uniform weights appears to perform the best among several multi-dimensional integration techniques. Random (Monte Carlo) approaches to approximate the integral are also possible: given a collection of grid points $\mathcal{S} = \{\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_m\}$, we approximate the integral as $I_{\mathcal{S}}(\boldsymbol{\phi}) := (\Delta/m) \sum_{\ell=1}^{m} \exp\{-\mathrm{cef}[\boldsymbol{\phi}](\boldsymbol{\xi}_\ell)\}$. This leads to an approximation of the objective f given by

$$f(\boldsymbol{\phi}) \approx \frac{1}{n}\mathbf{1}^\top \boldsymbol{\phi} + I_{\mathcal{S}}(\boldsymbol{\phi}) =: f_{\mathcal{S}}(\boldsymbol{\phi}). \tag{3.14}$$

Since $f_{\mathcal{S}}$ is a finite, convex function on $\mathbb{R}^n$, it has a subgradient at each $\boldsymbol{\phi} \in \mathbb{R}^n$, given by

$$\boldsymbol{g}_{\mathcal{S}}(\boldsymbol{\phi}) := \frac{1}{m} \sum_{\ell=1}^{m} \boldsymbol{G}(\boldsymbol{\phi}, \boldsymbol{\xi}_\ell).$$

As the effective domain of $\mathrm{cef}[\boldsymbol{\phi}](\cdot)$ is C_n, we consider grid points $\mathcal{S} \subseteq C_n$.

We now illustrate how these ideas allow us to approximate the gradient of the smoothing sequence, and initially consider Nesterov smoothing, with $\ell_u = \tilde{f}_u$. If $\mathcal{S} = \{\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_m\} \subseteq C_n$ denotes a collection of grid points (either deterministic or Monte Carlo based), then $\nabla_{\boldsymbol{\phi}} \ell_u$ can be approximated by $\tilde{\boldsymbol{g}}_{u,\mathcal{S}}$, where

$$\tilde{\boldsymbol{g}}_{u,\mathcal{S}}(\boldsymbol{\phi}) := \frac{1}{n}\mathbf{1} - \frac{\Delta}{m} \sum_{j=1}^{m} e^{-q_u[\boldsymbol{\phi}](\boldsymbol{\xi}_j)} \alpha_u^*[\boldsymbol{\phi}](\boldsymbol{\xi}_j). \tag{3.15}$$

In fact, we distinguish the cases of deterministic and random $\mathcal{S}$ by writing this approximation as $\tilde{\boldsymbol{g}}^{\mathrm{D}}_{u,\mathcal{S}}$ and $\tilde{\boldsymbol{g}}^{\mathrm{R}}_{u,\mathcal{S}}$ respectively.

[1] Once our algorithm terminates at $\tilde{\boldsymbol{\phi}}_T$, say, we evaluate the integral $I(\tilde{\boldsymbol{\phi}}_T)$ in the same way as [61]. Our final output, then is $\boldsymbol{\phi}_T := \tilde{\boldsymbol{\phi}}_T + \log I(\tilde{\boldsymbol{\phi}}_T)\mathbf{1}$; this final step not only improves the objective function, but also guarantees that $\exp[-\mathrm{cef}[\boldsymbol{\phi}_T](\cdot)]$ is a log-concave density.

[2] Yet another option involves sampling from a log-concave density [61, 63]. [10] discuss interesting polynomial-time sampling methods to approximate $I(\cdot)$, but as noted by [188], these methods may not be practically efficient, and we do not pursue them here.

For the randomized smoothing method with $\ell_u = \bar{f}_u$, the approximation is slightly more involved. Given m grid points $\mathcal{S} = \{\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_m\} \subseteq C_n$ (again either deterministic or random), and independent random vectors $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_m$, each uniformly distributed on the unit Euclidean ball in $\mathbb{R}^n$, we can approximate $\nabla_{\boldsymbol{\phi}} \ell_u(\boldsymbol{\phi})$ by

$$\bar{\boldsymbol{g}}^{\circ}_{u,\mathcal{S}}(\boldsymbol{\phi}) = \frac{1}{n}\mathbf{1} - \frac{\Delta}{m}\sum_{j=1}^{m} e^{-\mathrm{cef}[\boldsymbol{\phi}+u\boldsymbol{z}_j](\boldsymbol{\xi}_j)} \alpha^*[\boldsymbol{\phi}+u\boldsymbol{z}_j](\boldsymbol{\xi}_j), \tag{3.16}$$

with $\circ \in \{\mathrm{D}, \mathrm{R}\}$ again distinguishing the cases of deterministic and random $\mathcal{S}$.

3.3.2.2 Related literature

As mentioned above, Algorithm 3.1 is an accelerated version of the dual averaging method of [174], which to the best of our knowledge has not been studied in the context of log-concave density estimation previously. Nevertheless, related ideas have been considered for other optimization problems (e.g. [221, 75]). Relative to previous work, our approach is quite general, in that it applies to both of the smoothing techniques discussed in Section 3.3.1, and allows the use of both deterministic and random grids to approximate the gradients of the smoothing sequence. Another key difference with earlier work is that we allow the grid $\mathcal{S}$ to depend on t, so we write it as $\mathcal{S}_t$, with $m_t := |\mathcal{S}_t|$; in particular, inspired by both our theoretical results and numerical experience, we take (m_t) to be a suitable increasing sequence.

3.4 Theoretical analysis of optimization error of Algorithm 3.1

We have seen in Propositions 3.3 and 3.4 that the two smooth functions $\tilde{f}_u$ and $\bar{f}_u$ enjoy similar properties — according to Proposition 3.3(a) to (c) and Proposition 3.4(a) to (d), both $\tilde{f}_u$ and $\bar{f}_u$ satisfy the following assumption:

Assumption 3.1 (Assumptions on smoothing sequence)**.** *There exists $r \geq 0$ such that for any $\boldsymbol{\phi} \in \boldsymbol{\Phi}$,*

(a) we can find $B_0 > 0$ with $f(\boldsymbol{\phi}) \le \ell_u(\boldsymbol{\phi}) \le f(\boldsymbol{\phi}) + B_0 I(\boldsymbol{\phi})u$ for all $u \in [0, r]$;
(b) $\ell_{u'}(\boldsymbol{\phi}) \le \ell_u(\boldsymbol{\phi})$ for all $u' \in [0, u]$;
(c) for each $u \in [0, r]$, the function $\boldsymbol{\phi} \mapsto \ell_u(\boldsymbol{\phi})$ is convex and has B_1/u-Lipschitz gradient, for some $B_1 > 0$.

Recall from Section 3.3 that we have four possible choices corresponding to a combination of the smoothing and integral approximation methods, as summarized in Table 3.1.

Table 3.1: Summary of options for smoothing and gradient approximation methods.

Options	Smoothing	Approximation	Options	Smoothing	Approximation
1	$\breve{f}_u$	$\tilde{\boldsymbol{g}}^{\mathrm{D}}_{u,\mathcal{S}}$	3	$\bar{f}_u$	$\bar{\boldsymbol{g}}^{\mathrm{D}}_{u,\mathcal{S}}$
2	$\breve{f}_u$	$\tilde{\boldsymbol{g}}^{\mathrm{R}}_{u,\mathcal{S}}$	4	$\bar{f}_u$	$\bar{\boldsymbol{g}}^{\mathrm{R}}_{u,\mathcal{S}}$

Once we select an option, in line 3 of Algorithm 3.1, we can take

$$\boldsymbol{g}_t = \check{\boldsymbol{g}}^{\circ}_{u_t,\mathcal{S}_t}(\boldsymbol{\phi}_t^{(y)}),$$

where $\check{} \in \{\breve{}, \bar{}\}$ and $\circ \in \{\mathrm{D}, \mathrm{R}\}$. To encompass all four approximation choices in Line 3 of Algorithm 3.1, we make the following assumption on the gradient approximation error $\boldsymbol{e}_t := \boldsymbol{g}_t - \nabla_{\boldsymbol{\phi}} \ell_{u_t}(\boldsymbol{\phi}_t^{(y)})$:

Assumption 3.2 (Gradient approximation error)**.** *There exists $\sigma > 0$ such that*

$$\mathbb{E}\big(\|\boldsymbol{e}_t\|^2 | \mathcal{F}_{t-1}\big) \le \sigma^2/m_t \quad \text{for all } t \in \mathbb{N}_0, \tag{3.17}$$

where $\mathcal{F}_{t-1}$ denotes the σ-algebra generated by all random sources up to iteration $t-1$ (with $\mathcal{F}_{-1}$ denoting the trivial σ-algebra).

When $\mathcal{S}$ is a Monte Carlo random grid (options 2 and 4), the approximate gradient $\boldsymbol{g}_t$ is an average of m_t independent and identically distributed random vectors, each being an unbiased estimator of $\nabla \ell_{u_t}(\boldsymbol{\phi}_t^{(y)})$. Hence, (3.17) holds true with σ^2 determined by the bounds in Proposition 3.3(d) (option 2) and Proposition 3.4(e) (option 4). For a deterministic Riemann grid $\mathcal{S}$ and Nesterov's smoothing technique

(option 1), $\boldsymbol{e}_t$ is deterministic, and arises from using $\tilde{\boldsymbol{g}}_{u,\mathcal{S}}(\boldsymbol{\phi})$ in (3.15) to approximate $\nabla_{\boldsymbol{\phi}}\tilde{f}_u(\boldsymbol{\phi}) = \mathbb{E}[\tilde{\boldsymbol{G}}_u(\boldsymbol{\phi},\boldsymbol{\xi})]$. For the deterministic Riemann grid and randomized smoothing (option 3), the error $\boldsymbol{e}_t$ can be decomposed into a random estimation error term (induced by $\boldsymbol{z}_1,\ldots,\boldsymbol{z}_{m_t}$) and a deterministic approximation error term (induced by $\boldsymbol{\xi}_1,\ldots,\boldsymbol{\xi}_{m_t}$) as follows:

$$\begin{aligned}\boldsymbol{e}_t = \frac{1}{m_t}\sum_{j=1}^{m_t}\big(\boldsymbol{G}(\boldsymbol{\phi}_t^{(y)}+u_t\boldsymbol{z}_j,\boldsymbol{\xi}_j) - \mathbb{E}[\boldsymbol{G}(\boldsymbol{\phi}_t^{(y)}+u_t\boldsymbol{z},\boldsymbol{\xi}_j)|\mathcal{F}_{t-1}]\big)\\ + \bigg(\frac{1}{m_t}\sum_{j=1}^{m_t}\mathbb{E}[\boldsymbol{G}(\boldsymbol{\phi}_t^{(y)}+u_t\boldsymbol{z},\boldsymbol{\xi}_j)|\mathcal{F}_{t-1}] - \mathbb{E}[\boldsymbol{G}(\boldsymbol{\phi}_t^{(y)}+u_t\boldsymbol{z},\boldsymbol{\xi})|\mathcal{F}_{t-1}]\bigg).\end{aligned}$$

It can be shown using this decomposition that $\mathbb{E}(\|\boldsymbol{e}_t\|^2|\mathcal{F}_{t-1}) = O(1/m_t)$ under regularity conditions.

Theorem 3.1 below establishes our desired computational guarantees for Algorithm 3.1. We write $D := \sup_{\boldsymbol{\phi},\tilde{\boldsymbol{\phi}}\in\boldsymbol{\Phi}}\|\boldsymbol{\phi}-\tilde{\boldsymbol{\phi}}\|$ for the diameter of $\boldsymbol{\Phi}$.

Theorem 3.1. *Suppose that Assumptions 3.1 and 3.2 hold, and define the sequence* $(\theta_t)_{t\in\mathbb{N}_0}$ *by* $\theta_0 := 1$ *and* $\theta_{t+1} := 2\big(1+\sqrt{1+4/\theta_t^2}\big)^{-1}$ *for* $t\in\mathbb{N}_0$. *Let* $u>0$, *let* $u_t := \theta_t u$ *and take* $L_t = B_1/u_t$ *and* $\eta_t = \eta$ *for all* $t\in\mathbb{N}_0$ *as input parameters to Algorithm 3.1. Writing* $M_T^{(1)} := \sqrt{\sum_{t=0}^{T-1} m_t^{-1}}$ *and* $M_T^{(1/2)} := \sum_{t=0}^{T-1} m_t^{-1/2}$, *we have for any* $\boldsymbol{\phi}\in\boldsymbol{\Phi}$ *that*

$$\mathbb{E}[f(\boldsymbol{\phi}_T^{(x)})] - f(\boldsymbol{\phi}) \le \frac{B_1D^2}{Tu} + \frac{4B_0I(\boldsymbol{\phi})u}{T} + \frac{\eta D^2}{T} + \frac{\sigma^2(M_T^{(1)})^2}{T\eta} + \frac{2D\sigma M_T^{(1/2)}}{T}. \quad (3.18)$$

In particular, taking $\boldsymbol{\phi}=\boldsymbol{\phi}^*$, *and choosing* $u = (D/2)\sqrt{B_1/B_0}$ *and* $\eta = (\sigma M_T^{(1)})/D$, *we obtain*

$$\varepsilon_T := \mathbb{E}[f(\boldsymbol{\phi}_T^{(x)})] - f(\boldsymbol{\phi}^*) \le \frac{4\sqrt{B_0B_1}D}{T} + \frac{2\sigma D M_T^{(1)}}{T} + \frac{2D\sigma M_T^{(1/2)}}{T}. \quad (3.19)$$

Moreover, if we further assume that $\mathbb{E}(\boldsymbol{e}_t|\mathcal{F}_{t-1}) = \boldsymbol{0}$ *(e.g. by using options 2 and 4), then we can remove the last term of both inequalities above.*

For related general results that control the expected optimization error for

smoothed objective functions, see, e.g., [172], [207], [221], [75]. With deterministic grids (corresponding to options 1 and 3), if we take $|\mathcal{S}_t| = m$ for all t, then $M_T^{(1/2)} = T/\sqrt{m}$, and the upper bound in (3.19) does not converge to zero as $T \to \infty$. On the other hand, if we take $|\mathcal{S}_t| = t^2$, for example, then $\sup_{T\in\mathbb{N}} M_T^{(1)} < \infty$ and $M_T^{(1/2)} = \tilde{O}(1)$, and we find that $\varepsilon_T = \tilde{O}(1/T)$. For random grids (options 2 and 4), if we take $|\mathcal{S}_t| = m$ for all t, then $M_T^{(1)} = \sqrt{T/m}$ and we recover the $\varepsilon_T = O(1/\sqrt{T})$ rate for stochastic subgradient methods [185]. This can be improved to $\varepsilon_T = \tilde{O}(1/T)$ with $m_t = t$, or even $\varepsilon_T = O(1/T)$ if we choose $(m_t)_t$ such that $\sum_{t=0}^{\infty} m_t^{-1} < \infty$.

A direct application of the theory of [75] would yield an error rate of $\varepsilon_T = O(n^{1/4}/T + 1/\sqrt{T})$. On the other hand, Theorem 3.1 shows that, owing to the increasing sequence of grid sizes used to approximate the gradients in Step 3 of Algorithm 3.1, we can improve this rate to $\tilde{O}(1/T)$. Note however, that this improvement is in terms of the number of iterations T, and not the total number of stochastic oracle queries (equivalently, the total number of LPs (Q_0)), which is given by $T_{\text{query}} := \sum_{t=0}^{T-1} m_t$. [3] and [171] have shown that the optimal expected number of stochastic oracle queries is of order $1/\sqrt{T_{\text{query}}}$, which is attained by the algorithm of [75]. For our framework, by taking $m_t = t$, we have $T_{\text{query}} = \sum_{t=0}^{T-1} m_t = \tilde{O}(T^2)$, so after T_{query} stochastic oracle queries, our algorithm also attains the optimal error on the objective function scale, up to a logarithmic factor. Other advantages of our algorithm and the theoretical guarantees provided by Theorem 3.1 relative to the earlier contributions of [75] are that we do not require an upper bound on $I(\boldsymbol{\phi})$ and are able to provide a unified framework that includes Nesterov smoothing and an alternative gradient approximation approach by numerical integration in addition to randomized smoothing scheme with stochastic gradients. Moreover, we can exploit the specific structure of the log-concave density estimation problem to provide much better Lipschitz constants for the randomized smoothing sequence than would be obtained using the generic constants of [75]. For example, our upper bound in Proposition 3.4(a) is of order $O(n^{-1/2}\log^{1/2} n)$, whereas a naive application of the general theory of [75] would only yield a bound of $O(1)$. A further improvement in our bound comes from the fact that it now involves $I(\boldsymbol{\phi})$ directly, as opposed to an upper bound on this

quantity.

In Theorem 3.1, the computational guarantee depends upon B_0, B_1, σ in Assumptions 3.1 and 3.2. In light of Propositions 3.3 and 3.4, Table 3.2 illustrates how these quantities, and hence the corresponding guarantees, differ according to whether we use Nesterov smoothing or randomized smoothing.

The randomized smoothing procedure requires solving LPs, whereas Nesterov's smoothing technique requires solving QPs. While both of these problems are highly structured and can be solved efficiently by off-the-shelf solvers (e.g., [104]), we found the LP solution times to be faster than those for the QP. Additional computational details are discussed in Section 3.6.

Table 3.2: Comparion of constants in Assumption 3.1 for different smoothing schemes with $u \in [0, r]$. Here, σ corresponds to random grid points (options 2 and 4), the optimal η is taken to be proportional to σ, the optimal u is proportional to $\sqrt{B_1/B_0}$, we take $C_1 = \sqrt{\Delta e^{-\phi_0}}$; $\sqrt{B_0 B_1}$ determines the first term in the error rate.

	B_0	B_1	σ $(\eta \propto \sigma)$	u $(\propto \sqrt{B_1/B_0})$	$\sqrt{B_0 B_1}$
Nesterov	$1/2$	$\Delta e^{-\phi_0+r/2}(r+1)$	$\Delta e^{-\phi_0+r/2}$	$C_1 e^{r/4}\sqrt{(r+1)}O(1)$	
Randomized	$\sqrt{2n^{-1}\log n}e^r$	$\Delta e^{-\phi_0+r}\sqrt{n}$	$\Delta e^{-\phi_0+r}$	$C_1\tilde{O}(\sqrt{n})$	$C_1 e^r \tilde{O}(1)$

Note that Theorem 3.1 presents error bounds in expectation, though for option 1, since we use Nesterov's smoothing technique and the Riemann sum approximation of the integral, the guarantee in Theorem 3.1 holds without the need to take an expectation. Theorem 3.2 below presents corresponding high-probability guarantees. For simplicity, we present results for options 2 and 4, which rely on the following assumption:

Assumption 3.3. *Assume that* $\mathbb{E}(\boldsymbol{e}_t|\mathcal{F}_{t-1}) = \mathbf{0}$ *and that* $\mathbb{E}\big(e^{\|\boldsymbol{e}_t\|^2/\sigma_t^2} \mid \mathcal{F}_{t-1}\big) \leq e$, *where* $\sigma_t = \sigma/\sqrt{m_t}$.

Theorem 3.2. *Suppose that Assumptions 3.1 and 3.3 hold, and define the sequence* $(\theta_t)_{t\in\mathbb{N}_0}$ *by* $\theta_0 := 1$ *and* $\theta_{t+1} := 2\big(1 + \sqrt{1 + 4/\theta_t^2}\big)^{-1}$ *for* $t \in \mathbb{N}_0$. *Let* $u > 0$, *let* $u_t := \theta_t u$ *and take* $L_t = B_1/u_t$ *and* $\eta_t = \eta$ *for all* $t \in \mathbb{N}_0$ *as input parameters to Algorithm 3.1. Writing* $M_T^{(2)} := \sqrt{\sum_{t=0}^{T-1} m_t^{-2}}$ *and* $M_T^{(1)} := \sqrt{\sum_{t=0}^{T-1} m_t^{-1}}$, *and*

choosing $u = (D/2)\sqrt{B_1/B_0}$ *and* $\eta = (\sigma M_T^{(1)})/D$ *as in Theorem 3.1, for any* $\delta \in (0,1)$*, we have with probability at least* $1-\delta$ *that*

$$f(\boldsymbol{\phi}_T^{(x)}) - f(\boldsymbol{\phi}^*) \le \frac{2\sqrt{B_0 B_1} D}{T} + \frac{\sigma D M_T^{(1)}}{T} + \frac{4\sigma D M_T^{(1)} \sqrt{\log \frac{2}{\delta}}}{T} + \frac{4\sigma D \max\{ M_T^{(2)} \sqrt{2e \log \frac{2}{\delta}}, m_0^{-1} \log \frac{2}{\delta} \}}{M_T^{(1)} T}.$$

For option 3, we would need to consider the approximation error from the Riemann sum, and the final error rate would include additional $O(1/T)$ terms. We omit the details for brevity.

Finally in this section, we relate the error of the objective to the error in terms of $\boldsymbol{\phi}$, as measured through the squared L_2 distance between the corresponding lower convex envelope functions.

Theorem 3.3. *For any* $\boldsymbol{\phi} \in \boldsymbol{\Phi}$*, we have*

$$\int_{C_n} \{ \mathrm{cef}[\boldsymbol{\phi}](\boldsymbol{x}) - \mathrm{cef}[\boldsymbol{\phi}^*](\boldsymbol{x}) \}^2 \, d\boldsymbol{x} \le 2e^{\phi^0} \{ f(\boldsymbol{\phi}) - f(\boldsymbol{\phi}^*) \}. \tag{3.20}$$

3.5 Beyond log-concave density estimation

In this section, we extend our computational framework beyond the log-concave density family, through the notion of s-concave densities. For $s \in \mathbb{R}$, define domains $\mathcal{D}_s$ and $\psi_s : \mathcal{D}_s \to \mathbb{R}$ by

$$\mathcal{D}_s := \begin{cases} [0,\infty) & \text{if } s < 0 \\ (-\infty,\infty) & \text{if } s = 0 \\ (-\infty, 0] & \text{if } s > 0. \end{cases} \quad \text{and} \quad \psi_s(y) := \begin{cases} y^{1/s} & \text{if } s < 0 \\ e^{-y} & \text{if } s = 0 \\ (-y)^{1/s} & \text{if } s > 0. \end{cases}$$

Definition 3.1 (s-concave density, [201])**.** *For* $s \in \mathbb{R}$*, the class* $\mathcal{P}_s(\mathbb{R}^d)$ *of* s*-concave*

density functions on $\mathbb{R}^d$ *is given by*

$$\mathcal{P}_s(\mathbb{R}^d) := \left\{ p(\cdot) : p = \psi_s \circ \varphi \text{ for some } \varphi \in \mathcal{C}_d \text{ with } \mathrm{Im}(\varphi) \subseteq \mathcal{D}_s \cup \{\infty\}, \int_{\mathbb{R}^d} p = 1 \right\}.$$

For $s = 0$, the family of s-concave densities reduces to the family of log-concave densities. Moreover, for $s_1 < s_2$, we have $\mathcal{P}_{s_2}(\mathbb{R}^d) \subseteq \mathcal{P}_{s_1}(\mathbb{R}^d)$ [71, p. 86]. The s-concave density family introduces additional modelling flexibility, in particular allowing much heavier tails when $s < 0$ than the log-concave family, but we note that there is no guidance available in the literature on how to choose s.

For the problem of s-concave density estimation, we discuss two estimation methods, both of which have been previously considered in the literature, but for which there has been limited algorithmic development. The first is based on the maximum likelihood principle (Section 3.5.1), while the other is based on minimizing a Rényi divergence (Section 3.5.2).

3.5.1 Computation of the s-concave maximum likelihood estimator

[201] proved that a maximum likelihood estimator over $\mathcal{P}_s(\mathbb{R}^d)$ exists with probability one for $s \in (-1/d, \infty)$ and $n > \max\left(\frac{dr}{r-d}, d\right)$, where $r := -1/s$, and does not exist if $s < -1/d$. [73] provide some statistical properties of this estimator when $d = 1$. The maximum likelihood estimation problem is to compute

$$\hat{p}_n := \operatorname*{argmax}_{p \in \mathcal{P}_s(\mathbb{R}^d)} \sum_{i=1}^{n} \log p(\boldsymbol{x}_i), \tag{3.21}$$

or equivalently,

$$\operatorname*{argmax}_{\varphi \in \mathcal{C}_d : \mathrm{Im}(\varphi) \subseteq \mathcal{D}_s \cup \{\infty\}} \frac{1}{n} \sum_{i=1}^{n} \log \psi_s \circ \varphi(\boldsymbol{x}_i) \quad \text{subject to} \quad \int_{\mathbb{R}^d} \psi_s \circ \varphi(\boldsymbol{x}) \, d\boldsymbol{x} = 1. \tag{3.22}$$

We establish the following theorem:

Theorem 3.4. *Let $s \in [0,1]$ and suppose that the convex hull C_n of the data is d-dimensional (so that the s-concave MLE $\hat{p}_n$ exists and is unique). Then computing $\hat{p}_n$ in (3.21) is equivalent to the convex minimization problem of computing*

$$\boldsymbol{\phi}^* := \operatorname*{argmin}_{\boldsymbol{\phi}=(\phi_1,\ldots,\phi_n)\in\mathcal{D}_s^n} \left\{ -\frac{1}{n}\sum_{i=1}^{n} \log \psi_s(\phi_i) + \int_{C_n} \psi_s\big(\mathrm{cef}[\boldsymbol{\phi}](\boldsymbol{x})\big)\, d\boldsymbol{x} \right\}, \tag{3.23}$$

in the sense that $\hat{p}_n = \psi_s \circ \mathrm{cef}[\boldsymbol{\phi}^*]$.

Remark 3.1. *The equivalence result in Theorem 3.4 holds for any s (outside $[0,1]$) as long as the s-concave MLE exists. However, when $s \in [0,1]$, (3.23) is a convex optimization problem. The family of s-concave densities with $s < 0$ appears to be more useful from a statistical viewpoint as it allows for heavier tails than log-concave densities, but the MLE cannot be then computed via convex optimization. Nevertheless, the entropy minimization methods discussed in Section 3.5.2 can be used to obtain s-concave density estimates for $s > -1$.*

3.5.2 Quasi-concave density estimation

Another route to estimate an s-concave density (or even a more general class) is via the following problem:

$$\check{\varphi} := \operatorname*{argmin}_{\varphi\in\mathcal{C}_d:\mathrm{dom}(\varphi)=C_n} \left\{ \frac{1}{n}\sum_{i=1}^{n} \varphi(\boldsymbol{x}_i) + \int_{C_n} \Psi\big(\varphi(\boldsymbol{x})\big)\, d\boldsymbol{x} \right\}, \tag{3.24}$$

where $\Psi : \mathbb{R} \to (-\infty, \infty]$ is a decreasing, proper convex function. When $\Psi(y) = e^{-y}$, (3.24) is equivalent to the MLE for log-concave density estimation (3.1), by [61, Theorem 1]. This problem, proposed by [137], is called *quasi-concave* density estimation. [137, Theorem 4.1] show that under some assumptions on Ψ, there exists a solution to (3.24), and if Ψ is strictly convex, then the solution is unique. Furthermore, if Ψ is differentiable on the interior of its domain, then the optimal solution to the dual of (3.24) is a probability density p such that $p = -\Psi'(\varphi)$, and the dual problem can be regarded as minimizing different distances or entropies (depending on Ψ) between the empirical distribution of the data and p. In particular, when $\beta \geq 1$

and $\Psi(y) = \mathbb{1}_{\{y \le 0\}}(-y)^\beta/\beta$, and when $\beta < 0$ and $\Psi(y) = -y^\beta/\beta$ for $y \ge 0$ (with $\Psi(y) = \infty$ otherwise), the dual problem of (3.24) is essentially minimizing the Rényi divergence and we have the primal-dual relationship $p = |\varphi|^{\beta-1}$. In fact, this amounts to estimating an s-concave density via Rényi divergence minimization with $\beta = 1+1/s$ and $s \in (-1, \infty) \setminus \{0\}$. We therefore consider the problem

$$\min_{\substack{\varphi \in \mathcal{C}_d : \mathrm{dom}(\varphi) = C_n \\ \mathrm{Im}(\varphi) \subseteq \mathcal{D}_s}} \left\{ \frac{1}{n} \sum_{i=1}^n \varphi(\boldsymbol{x}_i) + \frac{1}{|1 + 1/s|} \int_{C_n} |\varphi(\boldsymbol{x})|^{1+1/s} \, d\boldsymbol{x} \right\}. \tag{3.25}$$

The proof of Theorem 3.5 is similar to that of Theorem 3.4, and is omitted for brevity.

Theorem 3.5. *Given a decreasing proper convex function Ψ, the quasi-concave density estimation problem* (3.24) *is equivalent to the following convex problem:*

$$\boldsymbol{\phi}^* := \operatorname*{argmin}_{\boldsymbol{\phi} \in \mathcal{D}_s^n} \left\{ \frac{1}{n} \mathbf{1}^\top \boldsymbol{\phi} + \int_{C_n} \Psi\big(\mathrm{cef}[\boldsymbol{\phi}](\boldsymbol{x})\big) \, d\boldsymbol{x} \right\}, \tag{3.26}$$

in the sense that $\check{\varphi} = \mathrm{cef}[\boldsymbol{\phi}^*]$, with corresponding density estimator $\tilde{p}_n = -\Psi' \circ \mathrm{cef}[\boldsymbol{\phi}^*]$.

The objective in (3.26) is convex, so our computational framework can be applied to solve this problem.

3.6 Computational experiments on simulated data

In this section, we present numerical experiments to study the different variants of our algorithm and compare them with existing methods based on convex optimization for the log-concave MLE. Our results are based on large-scale synthetic datasets with $n \in \{5{,}000, 10{,}000\}$ observations generated from standard d-dimensional normal and Laplace distributions with $d = 4$. Code for our experiments is available from the github repository `LogConcComp` available at:

`https://github.com/wenyuC94/LogConcComp`.

All computations were carried out on the MIT Supercloud Cluster [190] on an Intel Xeon Platinum 8260 machine, with 24 CPUs and 24GB of RAM. Our algorithms were written in Python; we used Gurobi [104] to solve the LPs and QPs.

Our first comparison method is that of [61], implemented in the `R` package `LogConcDEAD` [59], and denoted by CSS. The CSS algorithm terminates when $\|\boldsymbol{\phi}^{(t)} - \boldsymbol{\phi}^{(t-1)}\|_\infty \leq \tau$, and we consider $\tau \in \{10^{-2}, 10^{-3}, 10^{-4}\}$. Our other competing approach is the randomized smoothing method of [75], with random grids of a fixed grid size, which we denote here by RS-RF-m, with m being the grid size. To the best of our knowledge, this method has not been used to compute the log-concave MLE previously.

We denote the different variants of our algorithm as Alg-V, where Alg $\in$ {RS,NS} represents Algorithm 3.1 with Randomized smoothing and Nesterov smoothing, and $V \in \{\mathrm{DI}, \mathrm{RI}\}$ represents whether we use deterministic or random grids of increasing grid sizes to approximate the gradient. Further details of our input parameters are given in Appendix 3.A.3.

Figure 3-2 presents the relative objective error, defined for an algorithm with iterates $\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_t$ as

$$\texttt{relobj}(t) := \left| \frac{\min_{s \in [t]} f(\boldsymbol{\phi}_s) - f(\boldsymbol{\phi}^*)}{f(\boldsymbol{\phi}^*)} \right|, \tag{3.27}$$

against time (in minutes) and number of iterations. In the definition of the relative objective error in (3.27) above, $\boldsymbol{\phi}^*$ is taken as the CSS solution with tolerance $\tau = 10^{-4}$. The figure shows that randomized smoothing appears to outperform Nesterov smoothing in terms of the time taken to reach a desired relative objective error, since the former solves an LP (Q_0), whereas the latter has to solve a QP (Q_u); the number of iterations taken by the different methods is, however, similar. There is no clear winner between randomized and deterministic grids, and both appear to perform well.

Table 3.3 compares our proposed methods against the CSS solutions with different tolerances, in terms of running time, final objective function, and distances of the algorithm outputs to the optimal solution $\boldsymbol{\phi}^*$ and the truth $\boldsymbol{\phi}^{\mathrm{truth}}$. We find that all

of our proposals yield marked improvements in running time compared with the CSS solution: with $n = 10{,}000$, $d = 4$ and $\tau = 10^{-4}$, CSS takes more than 20 hours for all of the data sets we considered, whereas the RS-DI variant is around 50 times faster. The CSS solution may have a slightly improved objective function value on termination, but as shown in Table 3.3, all of our algorithms achieve an optimization error that is small by comparison with the statistical error, and from a statistical perspective, there is no point in seeking to reduce the optimization error further than this. Table 3.5 shows that the distances $\|\boldsymbol{\phi}^* - \boldsymbol{\phi}^{\text{truth}}\|/n^{1/2}$ are well concentrated around their means (i.e. do not vary greatly over different random samples drawn from the underlying distribution), which provides further reassurance that our solutions are sufficiently accurate for practical purposes. On the other hand, the CSS solution with tolerance 10^{-3} is not always sufficiently reliable in terms of its statistical accuracy, e.g. for a Laplace distribution with $n = 5{,}000$. Our further experiments on real data sets reported in Appendix 3.A.4 provide qualitatively similar conclusions.

Finally, Figure 3-3 compares our proposed multistage increasing grid sizes (RS-DI/RS-RI) (see Tables 5 and 6) with the fixed grid size (RS-RF) proposed by [75], under the randomized smoothing setting. We see that the benefits of using the increasing grid sizes as described by our theory carry over to improved practical performance, both in terms of running time and number of iterations.

3.A Additional implementational and experimental details

3.A.1 Initialization: non-convex method

[61] show that the negative log-density $-\log \hat{p}_n(\cdot)$ of the log-concave MLE is a piecewise-affine convex function over its domain C_n. This allows us to parametrize these functions as $\varphi(\boldsymbol{x}) := \max_{j\in[m]}\{\boldsymbol{a}_j^\top \boldsymbol{x} + b_j\}$ for $x \in C_n$, where $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_m \in \mathbb{R}^d$ and $b_1, \ldots, b_m \in \mathbb{R}$. We can then reformulate the problem as

Table 3.3: Comparison of our proposed methods with the CSS solution [61] and RS-RF [75]. On a single dataset, we ran 5 repetitions of each algorithm with different random seeds and report the median statistics. Here, `obj` and `relobj` denote the objective and relative objective error, respectively, `runtime` denotes the running time (in minutes), `dopt` and `dtruth` denote the (Euclidean) distances between the algorithm outputs and the optimal solution and the truth, respectively, `iter` denotes the number of iterations, `t0` denotes the total number of oracles (grid points), `a0` denotes the average number of oracles (grid points) per iteration, and `h0` denotes the harmonic average of grid sizes (which equals $T/(M_T^{(1)})^2$). For CSS, `param` is the tolerance τ; for RS-RF, `param` is the (fixed) grid size m. Here '-' means the running time of the corresponding algorithm exceeded 20 hours.

Normal, $n = 5{,}000, d = 4$

algo	param	obj	relobj	runtime	dopt	dtruth	iter	t0	a0	h0
CSS	1e-2	6.5209	1.1e-03	10.15	0.1955	0.2788				
	1e-3	6.5146	9.8e-05	110.04	0.0612	0.2465				
	1e-4	6.5140	0.0e-00	829.55	0.0000	0.2454				
RS-DI	None	6.5144	7.0e-05	16.04	0.0227	0.2499	128	6.18M	48.31K	20.23K
RS-RI	None	6.5150	1.6e-04	31.05	0.0289	0.2502	128	6.88M	53.75K	32.00K
NS-DI	None	6.5144	7.1e-05	89.94	0.0259	0.2497	128	6.18M	48.31K	20.23K
NS-RI	None	6.5149	1.5e-04	102.23	0.0312	0.2502	128	6.88M	53.75K	32.00K
RS-RF	5000	6.5174	5.2e-04	30.22	0.0575	0.2552	1024	5.12M	5.00K	5.00K
	10000	6.5168	4.4e-04	25.48	0.0429	0.2508	512	5.12M	10.00K	10.00K
	20000	6.5164	3.7e-04	23.05	0.0552	0.2567	256	5.12M	20.00K	20.00K
	40000	6.5158	2.9e-04	21.31	0.0344	0.2496	128	5.12M	40.00K	40.00K
	80000	6.5150	1.6e-04	42.25	0.0288	0.2499	128	10.24M	80.00K	80.00K

Laplace, $n = 5{,}000, d = 4$

algo	param	obj	relobj	runtime	dopt	dtruth	iter	t0	a0	h0
CSS	1e-2	7.9183	3.0e-02	30.77	2.5100	2.5449				
	1e-3	7.6994	1.1e-03	387.34	0.5985	0.6514				
	1e-4	7.6908	0.0e-00	1007.80	0.0000	0.2552				
RS-DI	None	7.6988	1.0e-03	17.64	0.0592	0.2304	128	7.24M	56.54K	34.84K
RS-RI	None	7.6939	4.0e-04	31.32	0.0424	0.2632	128	6.88M	53.75K	32.00K
NS-DI	None	7.6989	1.0e-03	109.68	0.0640	0.2259	128	7.24M	56.54K	34.84K
NS-RI	None	7.6943	4.5e-04	107.57	0.0362	0.2601	128	6.88M	53.75K	32.00K
RS-RF	5000	7.7048	1.8e-03	31.43	0.0628	0.2732	1024	5.12M	5.00K	5.00K
	10000	7.7059	2.0e-03	27.33	0.0621	0.2745	512	5.12M	10.00K	10.00K
	20000	7.6986	1.0e-03	24.53	0.0527	0.2696	256	5.12M	20.00K	20.00K
	40000	7.6979	9.2e-04	22.68	0.0852	0.2776	128	5.12M	40.00K	40.00K
	80000	7.6949	5.4e-04	43.27	0.0349	0.2614	128	10.24M	80.00K	80.00K

$\min_{\boldsymbol{a}_1,\ldots,\boldsymbol{a}_m\in\mathbb{R}^d, b_1,\ldots,b_m\in\mathbb{R}} f_0(\boldsymbol{a},\boldsymbol{b})$, with non-convex objective

$$f_0(\boldsymbol{a},\boldsymbol{b}) := \frac{1}{n}\sum_{i=1}^{n}\max_{j\in[m]}\{\boldsymbol{a}_j^\top \boldsymbol{x}_i + b_j\} + \int_{C_n} e^{-\max_{j\in[m]}\{\boldsymbol{a}_j^\top \boldsymbol{x}+b_j\}}\, d\boldsymbol{x}. \tag{3.28}$$

To approximate the integral, we use the same simple Riemann grid points mentioned in Section 3.3.2.1. Subgradients of the objective (3.28) are straightforward to compute

Table 3.4: Comparison of our proposed methods with the CSS solution [61] and RS-RF [75], but with $n = 10{,}000$. Details are given in the caption of Table 3.4.

Normal, $n = 10{,}000, d = 4$

algo	param	obj	relobj	runtime	dopt	dtruth	iter	t0	a0	h0
CSS	1e-2	6.5634	4.1e-04	24.72	0.1018	0.1911				
	1e-3	6.5612	7.3e-05	181.01	0.0462	0.1854				
	1e-4	6.5607	0.0e-00	-	0.0000	0.1859				
RS-DI	None	6.5621	2.1e-04	35.40	0.0411	0.1939	128	6.67M	52.14K	21.85K
RS-RI	None	6.5626	3.0e-04	65.18	0.0443	0.1939	128	6.88M	53.75K	32.00K
NS-DI	None	6.5620	2.0e-04	207.32	0.0429	0.1953	128	6.67M	52.14K	21.85K
NS-RI	None	6.5625	2.8e-04	215.51	0.0452	0.1959	128	6.88M	53.75K	32.00K
RS-RF	5000	6.5690	1.3e-03	64.80	0.1097	0.2205	1024	5.12M	5.00K	5.00K
	10000	6.5704	1.5e-03	56.26	0.0470	0.1854	512	5.12M	10.00K	10.00K
	20000	6.5656	7.5e-04	50.17	0.0412	0.1890	256	5.12M	20.00K	20.00K
	40000	6.5638	4.7e-04	46.24	0.0478	0.1948	128	5.12M	40.00K	40.00K
	80000	6.5627	3.0e-04	89.52	0.0446	0.1948	128	10.24M	80.00K	80.00K

Laplace, $n = 10{,}000, d = 4$

algo	param	obj	relobj	runtime	dopt	dtruth	iter	t0	a0	h0
CSS	1e-2	8.1796	5.8e-02	57.46	3.9044	3.9328				
	1e-3	7.7327	4.6e-04	-	0.3470	0.4081				
	1e-4	7.7292	0.0e-00	-	0.0000	0.2025				
RS-DI	None	7.7401	1.4e-03	42.70	0.0886	0.1825	128	8.14M	63.60K	39.20K
RS-RI	None	7.7370	1.0e-03	65.67	0.0753	0.2295	128	6.88M	53.75K	32.00K
NS-DI	None	7.7399	1.4e-03	263.40	0.0801	0.1724	128	8.14M	63.60K	39.20K
NS-RI	None	7.7365	9.4e-04	225.80	0.0480	0.2159	128	6.88M	53.75K	32.00K
RS-RF	5000	7.7541	3.2e-03	64.93	0.1051	0.2499	1024	5.12M	5.00K	5.00K
	10000	7.7543	3.2e-03	57.87	0.0918	0.2435	512	5.12M	10.00K	10.00K
	20000	7.7468	2.3e-03	51.25	0.1157	0.2529	256	5.12M	20.00K	20.00K
	40000	7.7511	2.8e-03	46.31	0.0907	0.2423	128	5.12M	40.00K	40.00K
	80000	7.7378	1.1e-03	89.13	0.0529	0.2230	128	10.24M	80.00K	80.00K

via the chain rule and the subgradient of the maximum function (see, e.g., [26]). After standardizing each coordinate of our data to have mean zero and unit variance, which does not affect the final outcome due to the affine equivariance of the log-concave MLE [78, Remark 2.4], we generate $m = 10$ initializing hyperplanes from a standard $(d+1)$-dimensional Gaussian distribution. We then obtain the initializer for our main algorithm by applying a vanilla subgradient method to the objective (3.28) [203, 185] with stepsize $t^{-1/2}$ at the tth iteration, terminating when the difference in the objective function at successive iterations drops below 10^{-4}, or after 100 iterations, whichever is the sooner. This technique is related to the non-convex method for log-concave density estimation proposed by [188], who considered a smoothed version of (3.28). Our goal here is only to seek a good initializer rather than the global optimum, and we found that the approach described above was effective in this respect, as well as

Table 3.5: Statistics of the distance between the optimal solution and truth. For each type of data set, we drew 40 random samples of the sizes given, and computed the log-concave MLE by CSS with tolerance 10^{-4}.

	mean	std.error	min	25%	50%	75%	max
normal ($n = 5{,}000$)	0.2565	0.0093	0.2415	0.2480	0.2574	0.2629	0.2745
Laplace ($n = 5{,}000$)	0.2590	0.0114	0.2366	0.2508	0.2578	0.2676	0.2825

being faster to compute than the method of [188].

3.A.2 Final polishing step

As mentioned in Section 3.3.2.1, once our algorithm terminates at $\tilde{\boldsymbol{\phi}}_T$, say, we evaluate the integral $I(\tilde{\boldsymbol{\phi}}_T)$ in the same way as [61]. Our final output, then is $\boldsymbol{\phi}_T := \tilde{\boldsymbol{\phi}}_T + \log I(\tilde{\boldsymbol{\phi}}_T)\mathbf{1}$; this final step not only improves the objective function, but also guarantees that $\exp[-\mathrm{cef}[\boldsymbol{\phi}_T](\cdot)]$ is a log-concave density. This can be shown by following the same arguments as in Steps 2-3 in the proof of Theorem 3.4.

3.A.3 Input parameter settings

According to Theorem 3.1, we should take $u = \frac{D}{2}\sqrt{\frac{B_1}{B_0}}$. By Table 3.2, for randomized smoothing, this is approximately $\frac{D}{2}C_1\sqrt{n}$, where $C_1 = \sqrt{\Delta e^{-\phi_0}}$. In our experiments for randomized smoothing, we chose $u = Dn^{1/4}/2$, while for Nesterov smoothing, we chose $u = D/2$. According to Theorem 3.1, $\eta = \sigma M_T^{(1)}/D$, where we took $\sigma = 10^{-4}$ for RS-RI and RS-DI, and $\sigma = 10^{-3}$ for NS-RI and NS-DI. For the competing RS-RF-m method, we present the better of the results from $\sigma \in \{10^{-3}, 10^{-4}\}$.

To illustrate the increasing grid size strategy we take in the experiments, we first present in Table 3.6 some potential schemes to achieve the $\tilde{O}(1/T)$ error rate on the objective function scale. In our experiments, we used the multi-stage increasing grid size scheme with $C_1 = 8$ and $\rho_1 = 2$. For the random grid (RI), we take $C = 5{,}000$ and $\rho = 2$. For the deterministic grid (DI), we first choose an axis-aligned grid with $m_{0,t}$ points in each dimension that encloses the convex hull C_n of the data. Then m_t is the number of these grid points that fall within C_n. Table 3.7 provides an illustration of

Table 3.6: Examples of increasing grid size ($|\mathcal{S}_t| = m_t$) schemes to achieve $\tilde{O}(1/T)$ rate (i.e. $M_T^{(1/2)} = \tilde{O}(1)$ for deterministic $\mathcal{S}_t$ and $M_T^{(1)} = \tilde{O}(1)$ for random $\mathcal{S}_t$). Here, C and C_1 are positive constants. For the multi-stage scheme, $a \geq 1$ denotes the current stage number.

Schemes	Grid Sizes	$M_T^{(1/2)} = \tilde{O}(1)$	$M_T^{(1)} = \tilde{O}(1)$
Exponential	$m_t = C\rho^t$	$\rho > 1$	$\rho > 1$
Polynomial	$m_t = Ct^\beta$	$\beta \geq 2$	$\beta \geq 1$
Multi-stage	$m_t = C\rho^a$ for $t \in \left[\mathbb{1}_{\{a\geq 1\}} + C_1\rho_1^{a-1}\mathbb{1}_{\{a\geq 2\}}, C_1\rho_1^a\right]$	$\rho_1^2 \leq \rho$	$\rho_1 \leq \rho$

this multi-stage strategy used in the numerical experiments for a Laplace distribution with $n = 5{,}000$ and $d = 4$. Code for the other settings is available in the github repository `LogConcComp`.

Table 3.7: Summary of increasing grid size strategy (illustrated with $n = 5{,}000$ observations from a Laplace distribution in four dimensions). We take a four stage grid strategy and 128 iterations in total, with stage lengths shown in second line. For deterministic grids (denoted by DI), we use $m_{0,t}$ to determine the grid size (third line in the table), and the fourth line of the table is the corresponding grid size. For random grids (denoted by RI), the fifth line is the grid size of random sample.

Stage number a		1	2	3	4
Stage length		16	16	32	64
DI	$m_{0,t}$	18	22	26	30
DI	m_t	10,656	23,582	45,969	81,558
RI	m_t	10,000	20,000	40,000	80,000

3.A.4 Experimental results on real data sets

We provide additional simulation results on three real data sets:

- **Stock returns:** The Stock returns real data consist of daily returns of four stocks[3] over $n = 10{,}000$ randomly sampled days between 1970 and 2010, normalized so that each dimension has unit variance. The real data are available at `https://stooq.com/db/h/`.

[3]International Business Machines Corporation (IBM.US), JPMorgan Chase & Co. (JPM.US), Caterpillar Inc. (CAT.US), 3M Company (MMM.US)

- **Census:** The Census real data consist of percentages of the population of different age groups (18-24, 25-44, 45-64 and 65+) for n = 10,000 randomly sampled Census tracts based on the 2015-2019 5-year ACS (American Community Survery)[4], and the data are normalized so that each dimension has unit variance. The data and description are available at `https://www.census.gov/topics/research/guidance/planning-databases/2021.html`.

- **Gas turbine:** The Gas turbine real data consist of 4 sensor measures[5] aggregated over one hour from a gas turbine for n = 10,000 hours between 2011 and 2015, normalized so that each dimension has unit variance. The data are available at `https://archive.ics.uci.edu/ml/datasets/Gas+Turbine+CO+and+NOx+Emission+Data+Set`.

Table 3.8, Figure 3-4 and Figure 3-5 provide simulation results that correspond to those in Table 3.3, Figure 3-2 and Figure 3-3 respectively, but for three three real data sets. The table and figures reveal a qualitatively very similar story to that presented for the simulated data in Section 3.6: the main conclusion is that our randomized smoothing approaches are significantly more computationally efficient than both the Nesterov smoothing and CSS methods.

3.B Appendix: Proofs

3.B.1 Proofs of Propositions 3.1 and 3.2

The proof of Proposition 3.1 is adapted from the proof of [10, Lemma 2], which in turn is based on [47, Lemma 8].

Proof of Proposition 3.1. The proof has three parts.

[4]pct_Pop_18_24_ACS_15_19, pct_Pop_25_44_ACS_15_19, pct_Pop_45_64_ACS_15_19, pct_Pop_65plus_ACS_15_19

[5]Ambient temperature (AT), Ambient pressure (AP), Carbon monoxide (CO), Nitrogen oxides (NOx).

Table 3.8: Comparison of our proposed methods with the CSS solution [61] and RS-RF [75], but on 3 real datasets. Details are given in the caption of Table 3.3.

Stock returns, $n = 10{,}000, d = 4$

algo	param	obj	relobj	runtime	dopt	iter	t0	a0	h0
CSS	1e-2	6.3395	6.7e-02	315.19	5.4659				
	1e-3	5.9458	8.7e-04	-	0.5130				
	1e-4	5.9406	0.0e-00	-	0.0000				
RS-DI	None	5.9589	3.1e-03	38.47	0.1428	128	7.79M	60.86K	30.22K
RS-RI	None	5.9778	6.3e-03	59.65	0.2032	128	6.88M	53.75K	32.00K
NS-DI	None	5.9506	1.7e-03	254.91	0.0792	128	7.79M	60.86K	30.22K
NS-RI	None	5.9672	4.5e-03	228.82	0.1362	128	6.88M	53.75K	32.00K
RS-RF	5000	6.0003	1.0e-02	61.98	0.2015	1024	5.12M	5.00K	5.00K
	10000	5.9886	8.1e-03	54.20	0.2354	512	5.12M	10.00K	10.00K
	20000	5.9852	7.5e-03	49.16	0.2141	256	5.12M	20.00K	20.00K
	40000	5.9940	9.0e-03	46.25	0.3194	128	5.12M	40.00K	40.00K
	80000	5.9665	4.4e-03	84.59	0.1055	128	10.24M	80.00K	80.00K

Census, $n = 10{,}000, d = 4$

algo	param	obj	relobj	runtime	dopt	iter	t0	a0	h0
CSS	1e-2	5.4458	9.4e-03	71.36	0.9222				
	1e-3	5.3953	1.2e-05	812.49	0.0098				
	1e-4	5.3952	0.0e-00	-	0.0000				
RS-DI	None	5.3995	8.0e-04	31.97	0.0478	128	6.19M	48.33K	25.79K
RS-RI	None	5.4003	9.4e-04	63.32	0.0506	128	6.88M	53.75K	32.00K
NS-DI	None	5.3992	7.3e-04	199.74	0.0453	128	6.19M	48.33K	25.79K
NS-RI	None	5.3992	7.4e-04	223.34	0.0475	128	6.88M	53.75K	32.00K
RS-RF	5000	5.4100	2.7e-03	69.86	0.1047	1024	5.12M	5.00K	5.00K
	10000	5.4093	2.6e-03	60.79	0.0796	512	5.12M	10.00K	10.00K
	20000	5.4074	2.3e-03	55.89	0.1236	256	5.12M	20.00K	20.00K
	40000	5.4059	2.0e-03	49.04	0.0587	128	5.12M	40.00K	40.00K
	80000	5.3998	8.6e-04	94.51	0.0477	128	10.24M	80.00K	80.00K

Gas turbine, $n = 10{,}000, d = 4$

algo	param	obj	relobj	runtime	dopt	iter	t0	a0	h0
CSS	1e-2	5.5920	5.7e-03	95.59	0.5908				
	1e-3	5.5617	2.7e-04	-	0.0994				
	1e-4	5.5602	0.0e-00	-	0.0000				
RS-DI	None	5.5693	1.6e-03	34.14	0.0897	128	7.08M	55.28K	25.53K
RS-RI	None	5.5633	5.7e-04	61.90	0.0493	128	6.88M	53.75K	32.00K
NS-DI	None	5.5689	1.6e-03	230.47	0.0914	128	7.08M	55.28K	25.53K
NS-RI	None	5.5622	3.7e-04	224.88	0.0499	128	6.88M	53.75K	32.00K
RS-RF	5000	5.5694	1.7e-03	67.28	0.1111	1024	5.12M	5.00K	5.00K
	10000	5.5670	1.2e-03	61.22	0.0794	512	5.12M	10.00K	10.00K
	20000	5.5673	1.3e-03	53.08	0.0455	256	5.12M	20.00K	20.00K
	40000	5.5657	9.9e-04	47.80	0.0547	128	5.12M	40.00K	40.00K
	80000	5.5632	5.5e-04	92.44	0.0501	128	10.24M	80.00K	80.00K

Part 1. We first prove that $\phi_i^* \in [\phi_0, \phi^0]$ for all $i \in [n]$; or equivalently

$$\begin{aligned} &\log \hat{p}_n(\boldsymbol{x}_i) \geq -(n-1) - d(n-1)\log\big(2n + 2nd\log(2nd)\big) - \log\Delta \\ &\log \hat{p}_n(\boldsymbol{x}_i) \leq 1 + d\log\big(2n + 2nd\log(2nd)\big) - \log\Delta \end{aligned} \tag{3.29}$$

for all $i \in [n]$. Define

$$M := \sup_{\boldsymbol{x} \in \mathbb{R}^d} \hat{p}_n(\boldsymbol{x}) = \max_{i \in [n]} \hat{p}_n(\boldsymbol{x}_i) \quad \text{and} \quad R := \frac{\max_{i \in [n]} \hat{p}_n(\boldsymbol{x}_i)}{\min_{i \in [n]} \hat{p}_n(\boldsymbol{x}_i)}.$$

We proceed to obtain an upper bound on R. To this end, let $\bar{p}_n$ denote the uniform density over C_n. If $\hat{p}_n(\boldsymbol{x}_i) = \bar{p}_n(\boldsymbol{x}_i) = 1/\Delta$ for all i, then (3.29) holds. So we may assume that $\hat{p}_n \neq \bar{p}_n$, so that $R > 1$ and $M > 1/\Delta$. For a density p on $\mathbb{R}^d$ and for $t \in \mathbb{R}$, let $L_p(t) := \{\boldsymbol{x} \in \mathbb{R}^d : p(\boldsymbol{x}) \geq t\}$ denote the super-level set of p at height t. Since $\hat{p}_n$ is supported on C_n, and since $\hat{p}_n(\boldsymbol{x}) \geq \min_{i \in [n]} \hat{p}_n(\boldsymbol{x}_i) = M/R$ for $\boldsymbol{x} \in C_n$, it follows by [47, Lemma 8][6] that when $R \geq e$,

$$\Delta = \mathrm{vol}(C_n) \leq \mathrm{vol}\big(L_{\hat{p}_n}(M/R)\big) \leq \frac{e \log^d R}{M}. \tag{3.30}$$

On the other hand, since $\inf_{\boldsymbol{x} \in C_n} \hat{p}_n(\boldsymbol{x}) = M/R$, we have $(M/R) \cdot \Delta \leq 1$, so for $R < e$, we have $M \leq R/\Delta < e/\Delta$. We deduce that

$$M \leq \frac{e \log_+^d R}{\Delta}, \tag{3.31}$$

for all $R > 1$, where $\log_+(x) := 1 \vee \log x$. Now, by the optimality of $\hat{p}_n$, we have

$$\begin{aligned} n \log(1/\Delta) = \sum_{i=1}^n \log \bar{p}_n(\boldsymbol{x}_i) \leq \sum_{i=1}^n \log \hat{p}_n(\boldsymbol{x}_i) &\leq \min_{i \in [n]} \log \hat{p}_n(\boldsymbol{x}_i) + (n-1) \max_{i \in [n]} \log \hat{p}_n(\boldsymbol{x}_i) \\ &= \log(M/R) + (n-1) \log M, \end{aligned} \tag{3.32}$$

so that $R \leq (M\Delta)^n$. It follows that when $R \geq e$, we have from (3.31) and the fact that $\log y \leq y$ for $y > 0$ that

$$R = \frac{R^2}{R} \leq \frac{e^{2n} \log^{2nd} R}{R} \leq e^{2n} (2nd)^{2nd}. \tag{3.33}$$

[6] In fact, the factor of e is omitted in the statement of [47, Lemma 8], but one can see from the authors' inequalities (27) and (28) that it should be present.

Since (3.33) holds trivially when $R < e$, we may combine (3.33) with (3.31) to obtain

$$\log M \leq 1 + d \log\big(2n + 2nd \log(2nd)\big) - \log \Delta \tag{3.34}$$

Moreover, from (3.32) and (3.34), we also have

$$\begin{aligned}\log(M/R) &\geq -n \log \Delta - (n-1) \log M \\ &\geq -(n-1) - d(n-1) \log\big(2n + 2nd \log(2nd)\big) - \log \Delta,\end{aligned}$$

as required.

Part 2. Now we extend the above result to all $\boldsymbol{\phi} \in \mathbb{R}^n$ such that $I(\boldsymbol{\phi}) = 1$ and $f(\boldsymbol{\phi}) \leq f(\bar{\boldsymbol{\phi}})$, where $\bar{\boldsymbol{\phi}}$ is defined just after Proposition 3.1. The key observation here is that the proof of Part 1 applies to any density with log-likelihood at least that of the uniform distribution over C_n. In particular, for any $\boldsymbol{\phi}$ satisfying these conditions, the density $p \in \mathcal{F}_d$ given by $p(\boldsymbol{x}) = \exp\{-\mathrm{cef}[\boldsymbol{\phi}](\boldsymbol{x})\}$ has log-likelihood at least that of the uniform distribution over C_n, so

$$-\phi^0 \leq \min_{i \in [n]} \log p(\boldsymbol{x}_i) \leq \sup_{\boldsymbol{x} \in \mathbb{R}^d} \log p(\boldsymbol{x}) \leq -\phi_0,$$

as required.

Part 3. We now consider the case for a general $\boldsymbol{\phi} \in \mathbb{R}^n$ with $f(\boldsymbol{\phi}) \leq f(\bar{\boldsymbol{\phi}})$. Let $\tilde{\boldsymbol{\phi}} := \boldsymbol{\phi} + \log I(\boldsymbol{\phi})\mathbf{1}$, so that $I(\tilde{\boldsymbol{\phi}}) = 1$ and $\mathrm{cef}[\tilde{\boldsymbol{\phi}}](\cdot) = \mathrm{cef}[\boldsymbol{\phi}](\cdot) + \log(I(\boldsymbol{\phi}))$. Furthermore,

$$f(\tilde{\boldsymbol{\phi}}) = \frac{1}{n}\mathbf{1}^\top \boldsymbol{\phi} + \log I(\boldsymbol{\phi}) + 1 \leq \frac{1}{n}\mathbf{1}^\top \boldsymbol{\phi} + I(\boldsymbol{\phi}) = f(\boldsymbol{\phi}) \leq f(\bar{\boldsymbol{\phi}}).$$

The result therefore follows by Part 2. □

Proof of Proposition 3.2. Recall our notation from Section 3.2 that $\boldsymbol{\alpha}^*[\boldsymbol{\phi}](\boldsymbol{x}) \in A[\boldsymbol{\phi}](\boldsymbol{x})$ denotes a solution to (Q_0) at $\boldsymbol{x} \in C_n$. Recall further from (3.9) that any

subgradient $\boldsymbol{g}(\boldsymbol{\phi})$ of f at $\boldsymbol{\phi}$ is of the form

$$\boldsymbol{g}(\boldsymbol{\phi}) = \frac{1}{n}\mathbf{1} - \boldsymbol{\gamma},$$

where $\boldsymbol{\gamma} := \Delta\mathbb{E}\big[\boldsymbol{\alpha}^*[\boldsymbol{\phi}](\boldsymbol{\xi})\exp\{-\mathrm{cef}[\boldsymbol{\phi}](\boldsymbol{\xi})\}\big]$ and $\boldsymbol{\xi}$ is uniformly distributed on C_n. Since $\boldsymbol{\alpha}^*$ lies in the simplex $\{\boldsymbol{\alpha} \in \mathbb{R}^n : \boldsymbol{\alpha} \geq 0, \mathbf{1}^\top\boldsymbol{\alpha} = 1\}$, we have $\boldsymbol{\gamma} \geq 0$ and

$$\mathbf{1}^\top\boldsymbol{\gamma} = \Delta\mathbb{E}[\mathbf{1}^\top\alpha^*[\boldsymbol{\phi}](\boldsymbol{\xi})\exp\{-\mathrm{cef}[\boldsymbol{\phi}](\boldsymbol{\xi})\}] = \Delta\mathbb{E}[\exp\{-\mathrm{cef}[\boldsymbol{\phi}](\boldsymbol{\xi})\}] = I(\boldsymbol{\phi}).$$

In particular, $\|\boldsymbol{\gamma}\|_1 = I(\boldsymbol{\phi})$, so

$$\|\boldsymbol{g}(\boldsymbol{\phi})\|^2 = \frac{1}{n} - \frac{2}{n}\mathbf{1}^\top\boldsymbol{\gamma} + \|\boldsymbol{\gamma}\|^2 \leq \frac{1}{n} - \frac{2}{n}I(\boldsymbol{\phi}) + I(\boldsymbol{\phi})^2.$$

If $I(\boldsymbol{\phi}) \leq 1/2$, then $\|\boldsymbol{g}(\boldsymbol{\phi})\|^2 \leq 1/4 + 1/n$; if $I(\phi) > 1/2$, then $\|\boldsymbol{g}(\boldsymbol{\phi})\|^2 \leq I(\boldsymbol{\phi})^2$. Therefore, $\|\boldsymbol{g}(\boldsymbol{\phi})\|^2 \leq \max\big\{1/4 + 1/n, I(\boldsymbol{\phi})^2\big\}$. □

3.B.2 Proofs of Proposition 3.3 and Proposition 3.4

The proof of Proposition 3.3 is based on the following properties of the quadratic program $q_u[\boldsymbol{\phi}](\boldsymbol{x})$ defined in (Q_u), as well as its unique optimizer $\boldsymbol{\alpha}_u^*[\boldsymbol{\phi}](\boldsymbol{x})$:

Proposition 3.5. *For $\boldsymbol{\phi} \in \boldsymbol{\Phi}$ and $\boldsymbol{x} \in C_n$, we have*

(a) $\|\boldsymbol{\alpha}_u^*[\boldsymbol{\phi}](\boldsymbol{x})\| \leq 1$ for any $\boldsymbol{x} \in C_n$ and $\boldsymbol{\phi} \in \boldsymbol{\Phi}$;

(b) $q_{u'}[\boldsymbol{\phi}](\boldsymbol{x}) + (u' - u)/2 \leq q_u[\boldsymbol{\phi}](\boldsymbol{x}) \leq q_{u'}[\boldsymbol{\phi}](\boldsymbol{x})$ for $u' \in [0, u]$;

(c) $\phi_0 - \frac{u}{2} \leq q_u[\boldsymbol{\phi}](\boldsymbol{x}) \leq \phi^0$ for all $u \geq 0$, $\boldsymbol{\phi} \in \boldsymbol{\Phi}$ and $\boldsymbol{x} \in C_n$;

(d) $\|\boldsymbol{\alpha}_u^*[\tilde{\boldsymbol{\phi}}](\boldsymbol{x}) - \boldsymbol{\alpha}_u^*[\boldsymbol{\phi}](\boldsymbol{x})\| \leq (1/u)\|\tilde{\boldsymbol{\phi}} - \boldsymbol{\phi}\|$ for any $u > 0$, $\boldsymbol{\phi}, \tilde{\boldsymbol{\phi}} \in \boldsymbol{\Phi}$, and any $\boldsymbol{x} \in C_n$.

Proof. The proof exploits ideas from [172]. For (a), observe that $\boldsymbol{\alpha}_u^*[\boldsymbol{\phi}](\boldsymbol{x}) \in E(\boldsymbol{x}) \subseteq \big\{\boldsymbol{\alpha} \in \mathbb{R}^n : \mathbf{1}_n^\top\boldsymbol{\alpha} = 1, \boldsymbol{\alpha} \geq 0\big\}$, and this simplex is the convex hull of $n + 1$ points that all lie in the closed unit Euclidean ball in $\mathbb{R}^n$.

The lower bound in (b) follows immediately from the definition of the quadratic

program in (Q_u). For the upper bound, for $u' \in [0, u]$, we have

$$\begin{aligned}
&q_u[\boldsymbol{\phi}](\boldsymbol{x}) - q_{u'}[\boldsymbol{\phi}](\boldsymbol{x}) \\
&= (\boldsymbol{\alpha}_u^*)^\top \boldsymbol{\phi} + \frac{u}{2}\big\|\boldsymbol{\alpha}_u^* - \boldsymbol{\alpha}_0\big\|^2 - (\boldsymbol{\alpha}_{u'}^*)^\top \boldsymbol{\phi} - \frac{u'}{2}\big\|\boldsymbol{\alpha}_{u'}^* - \boldsymbol{\alpha}_0\big\|^2 + \frac{u'-u}{2} \\
&\leq \frac{u-u'}{2}\left(\big\|\boldsymbol{\alpha}_{u'}^* - \boldsymbol{\alpha}_0\big\|^2 - 1\right) \leq \frac{u-u'}{2}\left(1 - \frac{2}{n} + \frac{1}{n} - 1\right) \leq 0,
\end{aligned}$$

in which $\boldsymbol{\alpha}_u^*$ and $\boldsymbol{\alpha}_{u'}^*$ denote $\boldsymbol{\alpha}_u^*[\boldsymbol{\phi}](\boldsymbol{x})$ and $\boldsymbol{\alpha}_{u'}^*[\boldsymbol{\phi}](\boldsymbol{x})$, respectively.

(c) For all $u \geq 0$, $\boldsymbol{\phi} \in \boldsymbol{\Phi}$ and $\boldsymbol{x} \in C_n$, we have

$$\begin{aligned}
q_u[\boldsymbol{\phi}](\boldsymbol{x}) = \inf_{\boldsymbol{\alpha} \in E(\boldsymbol{x})}\left(\boldsymbol{\alpha}^\top \boldsymbol{\phi} + \frac{u}{2}\|\boldsymbol{\alpha} - \boldsymbol{\alpha}_0\|^2 - \frac{u}{2}\right) &\geq \inf_{\boldsymbol{\alpha} \in E(\boldsymbol{x})} \boldsymbol{\alpha}^\top \boldsymbol{\phi} - \frac{u}{2} \\
&\geq \phi_0 \inf_{\boldsymbol{\alpha} \in E(\boldsymbol{x})} \boldsymbol{\alpha}^\top \mathbf{1}_n - \frac{u}{2} = \phi_0 - \frac{u}{2}.
\end{aligned}$$

Similarly,

$$q_u[\boldsymbol{\phi}](\boldsymbol{x}) \leq \phi^0 \sup_{\boldsymbol{\alpha} \in E(\boldsymbol{x})} \boldsymbol{\alpha}^\top \mathbf{1}_n + \frac{u}{2} \sup_{\boldsymbol{\alpha} \in E(\boldsymbol{x})} \|\boldsymbol{\alpha} - \boldsymbol{\alpha}_0\|^2 - \frac{u}{2} \leq \phi^0.$$

(d) Observe that

$$\boldsymbol{\alpha}_u^*[\boldsymbol{\phi}](\boldsymbol{x}) = \operatorname*{argmin}_{\boldsymbol{\alpha} \in E(\boldsymbol{x})}\left(\boldsymbol{\alpha}^\top \boldsymbol{\phi} + \frac{u}{2}\|\boldsymbol{\alpha} - \boldsymbol{\alpha}_0\|^2\right) = \operatorname*{argmin}_{\boldsymbol{\alpha} \in E(\boldsymbol{x})} \left\|\boldsymbol{\alpha} - \left(\boldsymbol{\alpha}_0 - \frac{\boldsymbol{\phi}}{u}\right)\right\|^2,$$

so $\boldsymbol{\alpha}_u^*[\boldsymbol{\phi}](\boldsymbol{x})$ is the Euclidean projection of $\boldsymbol{\alpha}_0 - (\boldsymbol{\phi}/u)$ onto $E(\boldsymbol{x})$. Since this projection is an ℓ_2-contraction, we deduce that

$$\big\|\boldsymbol{\alpha}_u^*[\tilde{\boldsymbol{\phi}}](\boldsymbol{x}) - \boldsymbol{\alpha}_u^*[\boldsymbol{\phi}](\boldsymbol{x})\big\| \leq \left\|\boldsymbol{\alpha}_0 - \frac{\tilde{\boldsymbol{\phi}}}{u} - \left(\boldsymbol{\alpha}_0 - \frac{\boldsymbol{\phi}}{u}\right)\right\| = \frac{1}{u}\|\tilde{\boldsymbol{\phi}} - \boldsymbol{\phi}\|,$$

as required. □

Proof of Proposition 3.3. (a) For $u \geq 0$ and $\boldsymbol{\phi} \in \boldsymbol{\Phi}$, let

$$\tilde{I}_u(\boldsymbol{\phi}) := \int_{C_n} e^{-q_u[\boldsymbol{\phi}](\boldsymbol{x})}\, d\boldsymbol{x} = \Delta \mathbb{E}(e^{-q_u[\boldsymbol{\phi}](\boldsymbol{\xi})}),$$

where $\boldsymbol{\xi}$ is uniformly distributed on C_n, so that $\tilde{I}_0(\boldsymbol{\phi}) = I(\boldsymbol{\phi})$. By definition of $\tilde{f}_u$, we have for $u' \in [0, u]$ that

$$\tilde{f}_u(\boldsymbol{\phi}) - \tilde{f}_{u'}(\boldsymbol{\phi}) = \tilde{I}_u(\boldsymbol{\phi}) - \tilde{I}_{u'}(\boldsymbol{\phi}) = \Delta\mathbb{E}\big(e^{-q_u[\boldsymbol{\phi}](\boldsymbol{\xi})} - e^{-q_{u'}[\boldsymbol{\phi}](\boldsymbol{\xi})}\big) \geq 0, \tag{3.35}$$

where the inequality follows from Proposition 3.5(b). Hence, for every $u \geq 0$ and $\boldsymbol{\phi} \in \boldsymbol{\Phi}$,

$$\tilde{I}_u(\boldsymbol{\phi}) \leq e^{u/2}\tilde{I}_0(\boldsymbol{\phi}) = e^{u/2} I(\boldsymbol{\phi}). \tag{3.36}$$

Now, from (3.35), Proposition 3.5(b) and (3.36), we deduce that

$$\tilde{f}_u(\boldsymbol{\phi}) - \tilde{f}_{u'}(\boldsymbol{\phi}) \leq \big(e^{(u-u')/2} - 1\big)\tilde{I}_{u'}(\boldsymbol{\phi}) \leq \frac{u-u'}{2} e^{u'/2} I(\boldsymbol{\phi}),$$

as required.

(b) For each $\boldsymbol{x} \in C_n$, the function $\boldsymbol{\phi} \mapsto q_u[\boldsymbol{\phi}](\boldsymbol{x})$ is the infimum of a set of affine functions of $\boldsymbol{\phi}$, so it is concave. Moreover, $y \mapsto e^{-y}$ is a decreasing convex function, so $\boldsymbol{\phi} \mapsto e^{-q_u[\boldsymbol{\phi}](\boldsymbol{x})}$ is convex, and it follows that $\boldsymbol{\phi} \mapsto (1/n)\mathbf{1}^\top\boldsymbol{\phi} + \Delta\mathbb{E}(e^{-q_u[\boldsymbol{\phi}](\boldsymbol{\xi})}) = \tilde{f}_u(\boldsymbol{\phi})$ is convex. Similarly to the proof of Proposition 3.2, any subgradient $\tilde{\boldsymbol{g}}_u(\phi)$ of $\tilde{f}_u$ at $\boldsymbol{\phi}$ satisfies

$$\|\tilde{\boldsymbol{g}}_u(\boldsymbol{\phi})\|^2 \leq \max\left\{\frac{1}{4} + \frac{1}{n}, \tilde{I}_u(\boldsymbol{\phi})^2\right\} \leq \max\left\{\frac{1}{4} + \frac{1}{n}, \Delta^2 e^{-2\phi_0+u}\right\}. \tag{3.37}$$

But $\boldsymbol{\phi}^* \in \boldsymbol{\Phi}$, so $\Delta^2 e^{-2\phi_0+u} \geq (\Delta e^{-\phi_0})^2 \geq I(\boldsymbol{\phi}^*)^2 = 1 \geq 1/4 + 1/n$. Hence, $\tilde{f}_u$ is $e^{-\phi_0+u/2}$-Lipschitz.

(c) To establish the Lipschitz property of $\nabla_{\boldsymbol{\phi}}\tilde{f}_u$, for any $\boldsymbol{x} \in C_n$, any $\boldsymbol{\phi}, \tilde{\boldsymbol{\phi}} \in \boldsymbol{\Phi}$, and $t \in [0, 1]$, we define

$$\eta(t) := e^{-q_u[\boldsymbol{\phi}+t(\tilde{\boldsymbol{\phi}}-\boldsymbol{\phi})](\boldsymbol{x})}.$$

Then

$$\eta'(t) = -e^{-q_u[\boldsymbol{\phi}+t(\tilde{\boldsymbol{\phi}}-\boldsymbol{\phi})](\boldsymbol{x})}(\tilde{\boldsymbol{\phi}} - \boldsymbol{\phi})^\top \boldsymbol{\alpha}_u^*[\boldsymbol{\phi} + t(\tilde{\boldsymbol{\phi}} - \boldsymbol{\phi})](\boldsymbol{x}).$$

By the mean value theorem there exists $t_0 \in [0,1]$ such that

$$\begin{aligned}
&\big|e^{-q_u[\tilde{\boldsymbol{\phi}}](\boldsymbol{x})} - e^{-q_u[\boldsymbol{\phi}](\boldsymbol{x})}\big| = |\eta(1)-\eta(0)| = |\eta'(t_0)| \\
&\leq e^{-q_u[\boldsymbol{\phi}+t_0(\tilde{\boldsymbol{\phi}}-\boldsymbol{\phi})](\boldsymbol{x})}\|\tilde{\boldsymbol{\phi}}-\boldsymbol{\phi}\|\big\|\boldsymbol{\alpha}_u^*[\boldsymbol{\phi}+t_0(\tilde{\boldsymbol{\phi}}-\boldsymbol{\phi})](\boldsymbol{x})\big\| \\
&\leq e^{-\phi_0+u/2}\|\tilde{\boldsymbol{\phi}}-\boldsymbol{\phi}\|, \qquad (3.38)
\end{aligned}$$

where the final bound follows from Proposition 3.5(a) and (c). Now, for any $\boldsymbol{x} \in C_n$, we have by (3.38) as well as Proposition 3.5(a), (c) and (d) that

$$\begin{aligned}
&\big\|e^{-q_u[\tilde{\boldsymbol{\phi}}](\boldsymbol{x})}\boldsymbol{\alpha}_u^*[\tilde{\boldsymbol{\phi}}](\boldsymbol{x}) - e^{-q_u[\boldsymbol{\phi}](\boldsymbol{x})}\boldsymbol{\alpha}_u^*[\boldsymbol{\phi}](\boldsymbol{x})\big\| \\
&= \big\|\big(e^{-q_u[\tilde{\boldsymbol{\phi}}](\boldsymbol{x})} - e^{-q_u[\boldsymbol{\phi}](\boldsymbol{x})}\big)\boldsymbol{\alpha}_u^*[\tilde{\boldsymbol{\phi}}](\boldsymbol{x}) + e^{-q_u[\boldsymbol{\phi}](\boldsymbol{x})}\big(\boldsymbol{\alpha}_u^*[\tilde{\boldsymbol{\phi}}](\boldsymbol{x}) - \boldsymbol{\alpha}_u^*[\boldsymbol{\phi}](\boldsymbol{x})\big)\big\| \\
&\leq e^{-\phi_0+u/2}\|\tilde{\boldsymbol{\phi}}-\boldsymbol{\phi}\| + e^{+}u/2-\phi_0\frac{1}{u}\|\tilde{\boldsymbol{\phi}}-\boldsymbol{\phi}\| = e^{-\phi_0+u/2}(1+u^{-1})\|\tilde{\boldsymbol{\phi}}-\boldsymbol{\phi}\|.
\end{aligned}$$

It follows that for any $\boldsymbol{\phi}, \tilde{\boldsymbol{\phi}} \in \boldsymbol{\Phi}$, we have

$$\begin{aligned}
\big\|\nabla_{\tilde{\boldsymbol{\phi}}}\tilde{f}_u(\tilde{\boldsymbol{\phi}}) - \nabla_{\boldsymbol{\phi}}\tilde{f}_u(\boldsymbol{\phi})\big\| &\leq \Delta\mathbb{E}\big\|e^{-q_u[\tilde{\boldsymbol{\phi}}](\boldsymbol{\xi})}\boldsymbol{\alpha}_u^*[\tilde{\boldsymbol{\phi}}](\boldsymbol{\xi}) - e^{-q_u[\boldsymbol{\phi}](\boldsymbol{\xi})}\boldsymbol{\alpha}_u^*[\boldsymbol{\phi}](\boldsymbol{\xi})\big\| \\
&\leq \Delta e^{-\phi_0+u/2}(1+u^{-1})\|\tilde{\boldsymbol{\phi}}-\boldsymbol{\phi}\|,
\end{aligned}$$

as required.

(d) For $u \geq 0$ and $\boldsymbol{\phi} \in \boldsymbol{\Phi}$, it follows from Proposition 3.5(a) and (c) that

$$\begin{aligned}
\mathbb{E}\big(\|\tilde{\boldsymbol{G}}_u(\boldsymbol{\phi},\boldsymbol{\xi}) - \nabla\tilde{f}_u(\boldsymbol{\phi})\|^2\big) &= \mathbb{E}\big\|\Delta e^{-q_u[\boldsymbol{\phi}](\boldsymbol{\xi})}\boldsymbol{\alpha}_u^*[\boldsymbol{\phi}](\boldsymbol{\xi}) - \Delta\mathbb{E}\big(e^{-q_u[\boldsymbol{\phi}](\boldsymbol{\xi})}\boldsymbol{\alpha}_u^*[\boldsymbol{\phi}](\boldsymbol{\xi})\big)\big\|^2 \\
&\leq \Delta^2\mathbb{E}\big\|e^{-q_u[\boldsymbol{\phi}](\boldsymbol{\xi})}\boldsymbol{\alpha}_u^*[\boldsymbol{\phi}](\boldsymbol{\xi})\big\|^2 \leq (\Delta e^{-\phi_0+u/2})^2,
\end{aligned}$$

as required. □

Proposition 3.6. *If $\boldsymbol{z}$ is uniformly distributed on the unit ℓ_2-ball in $\mathbb{R}^n$, then $\mathbb{E}(\|\boldsymbol{z}\|_\infty) \leq \sqrt{\frac{2\log n}{n+1}}$.*

Proof. By [223, Proposition 3], we have that $\boldsymbol{z} \stackrel{d}{=} U^{1/n}\boldsymbol{z}'$, where $U \sim \mathcal{U}[0,1]$, where $\boldsymbol{z}'$ is uniformly distributed on the unit sphere in $\mathbb{R}^n$, and where U and $\boldsymbol{z}'$ are independent.

Thus,

$$\mathbb{E}(\|\boldsymbol{z}\|_\infty) = \mathbb{E}(U^{1/n})\mathbb{E}(\|\boldsymbol{z}'\|_\infty) = \frac{n}{n+1}\mathbb{E}(\|\boldsymbol{z}'\|_\infty). \tag{3.39}$$

Moreover, if $\boldsymbol{\zeta} \sim \mathcal{N}(0, I_n)$, then $\|\boldsymbol{\zeta}\|$ and $\boldsymbol{\zeta}/\|\boldsymbol{\zeta}\|$ are independent, and $\boldsymbol{z}' \stackrel{d}{=} \boldsymbol{\zeta}/\|\boldsymbol{\zeta}\|$. It follows that

$$\mathbb{E}(\|\boldsymbol{z}'\|_\infty) = \mathbb{E}\left(\frac{\|\boldsymbol{\zeta}\|_\infty}{\|\boldsymbol{\zeta}\|}\right) \cdot \frac{\mathbb{E}(\|\boldsymbol{\zeta}\|)}{\mathbb{E}(\|\boldsymbol{\zeta}\|)} = \frac{\mathbb{E}(\|\boldsymbol{\zeta}\|_\infty)}{\mathbb{E}(\|\boldsymbol{\zeta}\|)} = \frac{\sqrt{2\log n} \cdot \Gamma(n/2)}{2^{1/2}\Gamma\big((n+1)/2\big)} \leq \frac{1}{n}\sqrt{2(n+1)\log n}, \tag{3.40}$$

where the final bound follows from bounds on the gamma function, e.g. [79, Lemma 12]. The result follows from (3.39) and (3.40). □

Proof of Proposition 3.4. (a) By Jensen's inequality,

$$\bar{f}_u(\boldsymbol{\phi}) = \mathbb{E}f(\boldsymbol{\phi} + u\boldsymbol{z}) \geq f(\boldsymbol{\phi}). \tag{3.41}$$

For the upper bound, let $\boldsymbol{v} \in \mathbb{R}^n$ have $\|\boldsymbol{v}\| \leq 1$ and, for some $\boldsymbol{\phi} \in \boldsymbol{\Phi}$, let $\bar{\boldsymbol{\phi}} := \boldsymbol{\phi} + u\boldsymbol{v}$. For any $\boldsymbol{\alpha} \in E(\boldsymbol{x})$, we have

$$|\boldsymbol{\alpha}^\top \bar{\boldsymbol{\phi}} - \boldsymbol{\alpha}^\top \boldsymbol{\phi}| = u|\boldsymbol{\alpha}^\top \boldsymbol{v}| \leq u\|\boldsymbol{\alpha}\|_1\|\boldsymbol{v}\|_\infty \leq u.$$

Therefore, for any $\boldsymbol{x} \in C_n$, we have $\mathrm{cef}[\bar{\boldsymbol{\phi}}](\boldsymbol{x}) \geq \mathrm{cef}[\boldsymbol{\phi}](\boldsymbol{x}) - u$. Hence

$$I(\bar{\boldsymbol{\phi}}) = \Delta\mathbb{E}[\exp\{-\mathrm{cef}[\bar{\boldsymbol{\phi}}](\boldsymbol{\xi})\}] \leq e^u \Delta\mathbb{E}[\exp\{-\mathrm{cef}[\boldsymbol{\phi}](\boldsymbol{\xi})\}] = e^u I(\boldsymbol{\phi}). \tag{3.42}$$

Recall that all subgradients of I at $\tilde{\boldsymbol{\phi}} \in \mathbb{R}^n$ are of the form $-\boldsymbol{\gamma}(\tilde{\boldsymbol{\phi}})$, where

$$\boldsymbol{\gamma}(\tilde{\boldsymbol{\phi}}) = \Delta\mathbb{E}\big(\boldsymbol{\alpha}\exp\{-\mathrm{cef}[\tilde{\boldsymbol{\phi}}](\boldsymbol{\xi})\}\big)$$

for some $\boldsymbol{\alpha} \in A[\tilde{\boldsymbol{\phi}}](\boldsymbol{x})$. Moreover, as we saw in the proof of Proposition 3.2, $\|\boldsymbol{\gamma}(\tilde{\boldsymbol{\phi}})\|_1 =$

$I(\tilde{\boldsymbol{\phi}})$. We deduce from [193, Theorem 24.7] that

$$\bar{f}_u(\boldsymbol{\phi}) - f(\boldsymbol{\phi}) = \mathbb{E}\big(I(\boldsymbol{\phi}+u\boldsymbol{z}) - I(\boldsymbol{\phi})\big) \le u \sup_{\boldsymbol{\phi}\in\boldsymbol{\Phi},\|\boldsymbol{v}\|\le 1} I(\boldsymbol{\phi}+u\boldsymbol{v})\mathbb{E}\big(\|\boldsymbol{z}\|_\infty\big)$$
$$\le I(\boldsymbol{\phi})ue^u\mathbb{E}\big(\|\boldsymbol{z}\|_\infty\big) \le I(\boldsymbol{\phi})ue^u\sqrt{\frac{2\log n}{n+1}},$$

where the final inequality uses Proposition 3.6.

(b) By the convexity of f, we have

$$\bar{f}_{u'}(\boldsymbol{\phi}) = \mathbb{E}\big(f(\boldsymbol{\phi}+u'\boldsymbol{z})\big) \le \frac{u'}{u}\bar{f}_u(\boldsymbol{\phi}) + \left(1-\frac{u'}{u}\right)f(\boldsymbol{\phi}) \le \bar{f}_u(\boldsymbol{\phi}),$$

where the last inequality uses property (a).

(c) For each $\boldsymbol{v}\in\mathbb{R}^n$ with $\|\boldsymbol{v}\|=1$, the map $\boldsymbol{\phi}\mapsto f(\boldsymbol{\phi}+u\boldsymbol{v})$ is convex, so $\boldsymbol{\phi}\mapsto \mathbb{E}\big(f(\boldsymbol{\phi}+u\boldsymbol{z})\big)=\bar{f}_u(\boldsymbol{\phi})$ is convex. The proof of the Lipschitz property is very similar to that of Proposition 3.3(b) and is omitted for brevity.

(d) As in the proof of (a), for any $\boldsymbol{v}\in\mathbb{R}^n$ with $\|\boldsymbol{v}\|\le 1$, $\boldsymbol{x}\in C_n$ and $\boldsymbol{\alpha}\in A[\boldsymbol{\phi}+u\boldsymbol{v}](\boldsymbol{x})$, we have

$$\big\|\boldsymbol{\alpha}e^{-\mathrm{cef}[\boldsymbol{\phi}+u\boldsymbol{v}](\boldsymbol{x})}\big\| \le e^{-\phi_0+u}.$$

Since $\nabla_{\boldsymbol{\phi}}\bar{f}_u(\boldsymbol{\phi}) = n^{-1}\mathbf{1} - \Delta\mathbb{E}\big(\boldsymbol{\alpha}^*[\boldsymbol{\phi}+u\boldsymbol{z}](\boldsymbol{\xi})e^{-\mathrm{cef}[\boldsymbol{\phi}+u\boldsymbol{z}](\boldsymbol{\xi})}\big)$, where $\boldsymbol{\alpha}^*[\boldsymbol{\phi}+u\boldsymbol{v}](\boldsymbol{x})\in A[\boldsymbol{\phi}+u\boldsymbol{v}](\boldsymbol{x})$, we have by [225, Lemma 8] that $\nabla_{\boldsymbol{\phi}}\bar{f}_u$ is $\Delta e^{-\phi_0+u}n^{1/2}/u$-Lipschitz.

(e) The proof is very similar to the proof of Proposition 3.3(d) and is omitted for brevity. □

3.B.3 Proofs of Theorem 3.1 and Theorem 3.2

We will make use of the following lemma:

Lemma 3.1 (Lemma 4.2 of [75]). *Let $\big(\ell_{u_t}(\boldsymbol{\phi})\big)_t$ be a smoothing sequence such that $\boldsymbol{\phi}\mapsto\ell_{u_t}(\boldsymbol{\phi})$ has L_t-Lipschitz gradient. Assume that $\ell_{u_t}(\boldsymbol{\phi})\le\ell_{u_{t-1}}(\boldsymbol{\phi})$ for $\boldsymbol{\phi}\in\boldsymbol{\Phi}$. Let $(\boldsymbol{\phi}_t^{(x)})_{t=0}^T, (\boldsymbol{\phi}_t^{(y)})_{t=0}^T, (\boldsymbol{\phi}_t^{(z)})_{t=0}^T$ be the sequences generated by Algorithm 3.1. Let $\boldsymbol{g}_t$ denote an approximation of $\nabla\ell_{u_t}(\boldsymbol{\phi}_t^{(y)})$ with error $\boldsymbol{e}_t=\boldsymbol{g}_t-\nabla\ell_{u_t}(\boldsymbol{\phi}_t^{(y)})$. Then for any*

$\boldsymbol{\phi} \in \boldsymbol{\Phi}$ *and* $t \in \mathbb{N}$*, we have*

$$\frac{1}{\theta_t^2}\ell_{u_t}(\boldsymbol{\phi}_{t+1}^{(x)}) \le \sum_{\tau=0}^{t}\frac{1}{\theta_\tau}\ell_{u_\tau}(\boldsymbol{\phi}) + \frac{1}{2}\left(L_{t+1} + \frac{\eta_{t+1}}{\theta_{t+1}}\right)\|\boldsymbol{\phi}-\boldsymbol{\phi}_0\|^2 + \sum_{\tau=0}^{t}\frac{\|\boldsymbol{e}_\tau\|^2}{2\theta_\tau\eta_\tau} + \sum_{\tau=0}^{t}\frac{1}{\theta_\tau}\langle \boldsymbol{e}_\tau, \boldsymbol{\phi}-\boldsymbol{\phi}_\tau^{(z)}\rangle.$$

Recall the definition of the diameter D of $\boldsymbol{\Phi}$ given just before Theorem 3.1.

Corollary 3.2. *Fix* $u, \eta > 0$*, and assume that Assumption 3.1 holds with* $r \ge u$*. Suppose in Algorithm 3.1 that* $u_t = \theta_t u$*,* $L_t = B_1/u_t$ *and* $\eta_t = \eta$*. Let* $(\boldsymbol{\phi}_t^{(x)})_{t=0}^T, (\boldsymbol{\phi}_t^{(y)})_{t=0}^T, (\boldsymbol{\phi}_t^{(z)})_{t=0}^T$ *be the sequences generated by Algorithm 3.1 and let* $\boldsymbol{e}_t = \boldsymbol{g}_t - \nabla\ell_{u_t}(\boldsymbol{\phi}_t^{(y)})$*. Then for any* $\boldsymbol{\phi} \in \boldsymbol{\Phi}$*, we have*

$$f(\boldsymbol{\phi}_T^{(x)}) - f(\boldsymbol{\phi}) \le \frac{B_1 D^2}{Tu} + \frac{\eta D^2}{T} + \frac{1}{T\eta}\sum_{t=0}^{T-1}\|\boldsymbol{e}_t\|^2 + \theta_{T-1}^2\sum_{t=0}^{T-1}\frac{1}{\theta_t}\langle \boldsymbol{e}_t, \boldsymbol{\phi}-\boldsymbol{\phi}_t^{(z)}\rangle + \frac{4B_0 I(\boldsymbol{\phi})u}{T}.$$

Proof. By induction, we have that $\theta_t \le 2/(t+2)$ and $\sum_{\tau=0}^t 1/\theta_\tau = 1/\theta_t^2$ for all $t = 0, 1, \ldots, T$ [207, 75]. Using Assumption 3.1, we have

$$\begin{aligned}\frac{1}{\theta_{T-1}^2}\big(f(\boldsymbol{\phi}_T^{(x)}) - f(\boldsymbol{\phi})\big) &\le \frac{1}{\theta_{T-1}^2}\ell_{u_{T-1}}(\boldsymbol{\phi}_T^{(x)}) - \sum_{t=0}^{T-1}\frac{1}{\theta_t}\big(\ell_{u_t}(\boldsymbol{\phi}) - B_0 u_t\big)\\ &= \frac{1}{\theta_{T-1}^2}\ell_{u_{T-1}}(\boldsymbol{\phi}_T^{(x)}) - \sum_{t=0}^{T-1}\frac{1}{\theta_t}\ell_{u_t}(\boldsymbol{\phi}) + TB_0 I(\boldsymbol{\phi})u.\end{aligned}$$

Hence, by Lemma 3.1,

$$
\begin{aligned}
&f(\boldsymbol{\phi}_T^{(x)}) - f(\boldsymbol{\phi}) \\
&\quad \le \frac{\theta_{T-1}^2}{2}\|\boldsymbol{\phi}-\boldsymbol{\phi}_0\|^2\bigg(L_T + \frac{\eta_T}{\theta_T}\bigg) + \sum_{t=0}^{T-1}\frac{\theta_{T-1}^2}{2\theta_t\eta_t}\|\boldsymbol{e}_t\|^2 \\
&\qquad\qquad + \theta_{T-1}^2\sum_{t=0}^{T-1}\frac{1}{\theta_t}\langle \boldsymbol{e}_t, \boldsymbol{\phi}-\boldsymbol{\phi}_t^{(z)}\rangle + \theta_{T-1}^2 T B_0 I(\boldsymbol{\phi})u \\
&\quad \le \frac{B_1 D^2\theta_{T-1}^2}{2u\theta_T} + \frac{\eta_T\theta_{T-1}^2 D^2}{2\theta_T} + \sum_{t=0}^{T-1}\frac{\theta_{T-1}^2}{2\theta_t\eta_t}\|\boldsymbol{e}_t\|^2 \\
&\qquad\qquad + \theta_{T-1}^2\sum_{t=0}^{T-1}\frac{1}{\theta_t}\langle \boldsymbol{e}_t, \boldsymbol{\phi}-\boldsymbol{\phi}_t^{(z)}\rangle + \theta_{T-1}^2 T B_0 I(\boldsymbol{\phi})u \\
&\quad \le \frac{B_1 D^2}{Tu} + \frac{\eta_T D^2}{T} + \frac{1}{T}\sum_{t=0}^{T-1}\frac{1}{\eta_t}\|\boldsymbol{e}_t\|^2 + \theta_{T-1}^2\sum_{t=0}^{T-1}\frac{1}{\theta_t}\langle \boldsymbol{e}_t, \boldsymbol{\phi}-\boldsymbol{\phi}_t^{(z)}\rangle + \frac{4B_0 I(\boldsymbol{\phi})u}{T},
\end{aligned}
$$

where we have used the facts that $\theta_{T-1}^2/\theta_T = \theta_T/(1-\theta_T) \le 2/T$ and $\theta_{T-1}^2/\theta_t \le \theta_{T-1} \le 2/T$ for $t \in \{0, 1, \ldots, T-1\}$ and $\theta_{T-1}^2 \le 4/T^2$. □

Proof of Theorem 3.1. According to Corollary 3.2, it suffices to bound $\theta_{T-1}^2\sum_{t=0}^{T-1}\frac{1}{\theta_t}\mathbb{E}\big(\langle \boldsymbol{e}_t, \boldsymbol{\phi}-\boldsymbol{\phi}_t^{(z)}\rangle\big)$ and $\sum_{t=0}^{T-1}\mathbb{E}\big(\|\boldsymbol{e}_t\|^2\big)$. To this end, we have by Assumption 3.2 that

$$
\begin{aligned}
\mathbb{E}\big(\langle \boldsymbol{e}_t, \boldsymbol{\phi}-\boldsymbol{\phi}_t^{(z)}\rangle\big) &= \mathbb{E}\big(\mathbb{E}(\langle \boldsymbol{e}_t, \boldsymbol{\phi}-\boldsymbol{\phi}_t^{(z)}\rangle \mid \mathcal{F}_{t-1})\big) \le \mathbb{E}\big(\mathbb{E}(\|\boldsymbol{e}_t\|\|\boldsymbol{\phi}-\boldsymbol{\phi}_t^{(z)}\| \mid \mathcal{F}_{t-1})\big) \\
&\le D\cdot\mathbb{E}\big(\mathbb{E}(\|\boldsymbol{e}_t\| \mid \mathcal{F}_{t-1})\big) \le D\cdot\mathbb{E}\Big(\sqrt{\mathbb{E}\big(\|\boldsymbol{e}_t\|^2 \mid \mathcal{F}_{t-1}\big)}\Big) \le \frac{D\sigma}{\sqrt{m_t}}.
\end{aligned} \tag{3.43}
$$

We deduce that

$$
\theta_{T-1}^2\sum_{t=0}^{T-1}\frac{1}{\theta_t}\mathbb{E}\big(\langle \boldsymbol{e}_t, \boldsymbol{\phi}-\boldsymbol{\phi}_t^{(z)}\rangle\big) \le \frac{2D\sigma M_T^{(1/2)}}{T}. \tag{3.44}
$$

Moreover, by Assumption 3.2 again,

$$
\sum_{t=0}^{T-1}\mathbb{E}\big(\|\boldsymbol{e}_t\|^2\big) \le \sigma^2\sum_{t=0}^{T-1}\frac{1}{m_t} = \sigma^2(M_T^{(1)})^2. \tag{3.45}
$$

The bound (3.18) follows from Corollary 3.2, together with (3.44) and (3.45), and the bound (3.19) then follows directly from the parameter choice of u and η and the fact that $I(\boldsymbol{\phi}^*) = 1$.

Finally, if $\mathbb{E}(\boldsymbol{e}_t \mid \mathcal{F}_{t-1}) = \mathbf{0}$, then

$$\mathbb{E}\big(\langle \boldsymbol{e}_t, \boldsymbol{\phi} - \boldsymbol{\phi}_t^{(z)} \rangle\big) = \mathbb{E}\big(\mathbb{E}(\langle \boldsymbol{e}_t, \boldsymbol{\phi} - \boldsymbol{\phi}_t^{(z)} \rangle \mid \mathcal{F}_{t-1})\big) = \mathbb{E}\big(\langle \mathbb{E}(\boldsymbol{e}_t | \mathcal{F}_{t-1}), \boldsymbol{\phi} - \boldsymbol{\phi}_t^{(z)} \rangle\big) = 0$$

where the second equality uses the fact that $\boldsymbol{\phi} - \boldsymbol{\phi}_t^{(z)}$ is $\mathcal{F}_{t-1}$-measurable. This allows us to remove the last term of the two inequalities in the theorem. □

Proof of Theorem 3.2. According to Corollary 3.2, it suffices to obtain a high-probability bound for $\theta_{T-1}^2 \sum_{t=0}^{T-1} \frac{1}{\theta_t} \langle \boldsymbol{e}_t, \boldsymbol{\phi} - \boldsymbol{\phi}_t^{(z)} \rangle$ and $\sum_{t=0}^{T-1} \|\boldsymbol{e}_t\|^2$. Writing $\boldsymbol{\zeta}_t := \boldsymbol{\phi} - \boldsymbol{\phi}_t^{(z)}$, we have from the proof of Theorem 3.1 that $(1/\theta_t)\langle \boldsymbol{e}_t, \boldsymbol{\zeta}_t \rangle$ is a martingale difference sequence under Assumption 3.3. Note that $\boldsymbol{\zeta}_t$ is $\mathcal{F}_{t-1}$-measurable, and we will now show that $\langle \boldsymbol{e}_t, \boldsymbol{\zeta}_t \rangle$ is $\sqrt{2}\sigma_t D$ sub-Gaussian, conditional on $\mathcal{F}_{t-1}$.

For any $x \in \mathbb{R}$, we have $e^x \le x + e^{x^2}$. Hence, for $\lambda \in \mathbb{R}$ such that $\lambda^2 \sigma_t^2 D^2 \le 1$, we have by the conditional version of Jensen's inequality that

$$\begin{aligned}
\mathbb{E}\big(e^{\lambda \langle \boldsymbol{e}_t, \boldsymbol{\zeta}_t \rangle} \mid \mathcal{F}_{t-1}\big) &\le \mathbb{E}\big(\lambda \langle \boldsymbol{e}_t, \boldsymbol{\zeta}_t \rangle \mid \mathcal{F}_{t-1}\big) + \mathbb{E}\big(e^{\lambda^2 \langle \boldsymbol{e}_t, \boldsymbol{\zeta}_t \rangle^2} \mid \mathcal{F}_{t-1}\big) \\
&\le \mathbb{E}\big(e^{\lambda^2 \|\boldsymbol{e}_t\|^2 D^2} | \mathcal{F}_{t-1}\big) \le e^{\lambda^2 \sigma_t^2 D^2}.
\end{aligned}$$

On the other hand, if $\lambda^2 \sigma_t^2 D^2 > 1$, then since $2ab \le a^2 + b^2$ for all $a, b \in \mathbb{R}$, we have

$$\begin{aligned}
\mathbb{E}\big(e^{\lambda \langle \boldsymbol{e}_t, \boldsymbol{\zeta}_t \rangle} \mid \mathcal{F}_{t-1}\big) &\le e^{\lambda^2 \sigma_t^2 D^2 / 2} \mathbb{E}\big(e^{\langle \boldsymbol{e}_t, \boldsymbol{\zeta}_t \rangle^2 / (2\sigma_t^2 D^2)} \mid \mathcal{F}_{t-1}\big) \\
&\le e^{\lambda^2 \sigma_t^2 D^2 / 2} \mathbb{E}\big(e^{\|\boldsymbol{e}_t\|^2 / (2\sigma_t^2)} \mid \mathcal{F}_{t-1}\big) \le e^{\lambda^2 \sigma_t^2 D^2 / 2} e^{1/2} \le e^{\lambda^2 \sigma_t^2 D^2}.
\end{aligned}$$

We deduce that $\langle \boldsymbol{e}_t, \boldsymbol{\zeta}_t \rangle / \theta_t$ is $(\sqrt{2}\sigma_t D)/\theta_t$ sub-Gaussian, conditional on $\mathcal{F}_{t-1}$. Applying

the Azuma–Hoeffding inequality (e.g. [12]) therefore yields that for every $\epsilon > 0$,

$$\mathbb{P}\bigg(\theta_{T-1}^2 \sum_{t=0}^{T-1} \frac{1}{\theta_t}\langle \boldsymbol{e}_t, \boldsymbol{\zeta}_t\rangle \geq \epsilon\bigg) \leq \exp\bigg(-\frac{\epsilon^2}{4D^2\theta_{T-1}^2 \sum_{t=0}^{T-1} \sigma_t^2 \theta_{T-1}^2/\theta_t^2}\bigg)$$
$$\leq \exp\bigg(-\frac{T^2\epsilon^2}{16D^2\sigma^2 (M_T^{(1)})^2}\bigg),$$

where the last inequality uses the facts that $\theta_{T-1} \leq \theta_t$ and $\theta_{T-1} \leq 2/T$. Therefore, for every $\delta \in (0,1)$, we have with probability at least $1 - \delta/2$ that

$$\theta_{T-1}^2 \sum_{t=0}^{T-1} \frac{1}{\theta_t}\langle \boldsymbol{e}_t, \boldsymbol{\zeta}_t\rangle \leq \frac{4\sigma D M_T^{(1)} \sqrt{\log(2/\delta)}}{T}. \tag{3.46}$$

Next we will turn to finding a tail bound for $\sum_{t=0}^{T-1} \|\boldsymbol{e}_t\|^2$. By Assumption 3.3 and Jensen's inequality, we have

$$\sum_{t=0}^{T-1} \mathbb{E}\big(\|\boldsymbol{e}_t\|^2 \;\big|\; \mathcal{F}_{t-1}\big) \leq \sum_{t=0}^{T-1} \sigma_t^2 \log\big(\mathbb{E}(e^{\|\boldsymbol{e}_t\|^2/\sigma_t^2} | \mathcal{F}_{t-1})\big) \leq \sigma^2 (M_T^{(1)})^2. \tag{3.47}$$

Now define the random variables $\Xi_t := \|\boldsymbol{e}_t\|^2 - \mathbb{E}\big(\|\boldsymbol{e}_t\|^2 \;\big|\; \mathcal{F}_{t-1}\big)$. Then by Markov's inequality, for every $\epsilon > 0$,

$$\mathbb{P}\big(\Xi_t > \epsilon \;\big|\; \mathcal{F}_{t-1}\big) \leq \mathbb{P}\big(\|\boldsymbol{e}_t\|^2/\sigma_t^2 > \epsilon/\sigma_t^2 \;\big|\; \mathcal{F}_{t-1}\big) \leq e^{-\epsilon/\sigma_t^2} \mathbb{E}\big(e^{\|\boldsymbol{e}_t\|^2/\sigma_t^2} \;\big|\; \mathcal{F}_{t-1}\big) \leq e^{1-\epsilon/\sigma_t^2}.$$

Moreover, by Markov's inequality again, and then Jensen's inequality, we have for every $\epsilon > 0$ that

$$\mathbb{P}\big(\Xi_t < -\epsilon \;\big|\; \mathcal{F}_{t-1}\big) \leq e^{-\epsilon/\sigma_t^2} \mathbb{E}\big(e^{\mathbb{E}(\|\boldsymbol{e}_t\|^2/\sigma_t^2 | \mathcal{F}_{t-1}) - \|\boldsymbol{e}_t\|^2/\sigma_t^2} \;\big|\; \mathcal{F}_{t-1}\big)$$
$$\leq e^{-\epsilon/\sigma_t^2} e^{\mathbb{E}(\|\boldsymbol{e}_t\|^2/\sigma_t^2 | \mathcal{F}_{t-1})} \leq e^{1-\epsilon/\sigma_t^2}.$$

It follows by, e.g., [75, Lemma F.7] that Ξ_t is sub-exponential with parameters $\lambda_t := 1/(2\sigma_t^2) = m_t/(2\sigma^2)$ and $\tau_t^2 := 16e\sigma_t^4 = 16e\sigma^4/m_t^2$, in the sense that

$$\mathbb{E}\big(e^{\lambda \Xi_t} \;\big|\; \mathcal{F}_{t-1}\big) \leq e^{8e\lambda^2\sigma_t^4}, \tag{3.48}$$

for $|\lambda| \leq 1/(2\sigma_t^2)$.

Now define $\Lambda_T := \min_{t=0,\dots,T-1} \lambda_t = m_0/(2\sigma^2)$ (as we assume (m_t) is increasing) and $C_T := \big(\sum_{t=0}^{T-1} \tau_t^2\big)^{1/2} = 4e^{1/2}\sigma^2 M_T^{(2)}$. We claim that $\sum_{t=0}^{T-1} \Xi_t$ is sub-exponential with parameters Λ_T and C_T, and prove this by induction on T. The base case $T = 1$ holds by (3.48), so suppose it holds for a given $T \in \mathbb{N}$. Then for $\lambda \in \mathbb{R}$ with $|\lambda| \leq \min(\Lambda_T, \lambda_T) = \Lambda_{T+1}$, we have

$$\begin{aligned}\mathbb{E}\Big\{\exp\Big(\lambda \sum_{t=0}^{T} \Xi_t\Big)\Big\} &= \mathbb{E}\Big[\exp\Big(\lambda \sum_{t=0}^{T-1} \Xi_t\Big)\mathbb{E}\big\{\exp\big(\lambda \Xi_T | \mathcal{F}_{T-1}\big)\big\}\Big] \\ &\leq e^{(\lambda^2 C_T^2 + 16e\lambda^2\sigma_T^4)/2} = e^{\lambda^2 C_{T+1}^2/2},\end{aligned}$$

which proves the claim by induction. We deduce by, e.g. [42, Lemma 1.4.1], that for every $\epsilon > 0$ and $T \in \mathbb{N}$,

$$\mathbb{P}\Big(\sum_{t=0}^{T-1} \Xi_t \geq \epsilon\Big) \leq \exp\Big(-\min\Big\{\frac{\epsilon^2}{2C_T^2}, \frac{\Lambda_T \epsilon}{2}\Big\}\Big).$$

In other words, with probability at least $1 - \delta/2$,

$$\sum_{t=0}^{T-1} \Xi_t \leq 4\sigma^2 \max\Big\{M_T^{(2)}\sqrt{2e\log\frac{2}{\delta}}, \frac{1}{m_0}\log\frac{2}{\delta}\Big\}. \tag{3.49}$$

Applying (3.46), (3.47) and (3.49) in Corollary 3.2, together with a union bound, yields that with probability at least $1 - \delta$,

$$\begin{aligned}f(\boldsymbol{\phi}_T^{(x)}) - f(\boldsymbol{\phi}) \leq \frac{B_1 D^2}{Tu} + \frac{4B_0 I(\boldsymbol{\phi})u}{T} + \frac{\eta D^2}{T} + \frac{\sigma^2 (M_T^{(1)})^2}{T\eta} + \frac{4\sigma D M_T^{(1)}\sqrt{\log(2/\delta)}}{T} \\ + \frac{4\sigma^2 \max\big\{M_T^{(2)}\sqrt{2e\log(2/\delta)}, m_0^{-1}\log(2/\delta)\big\}}{T\eta}.\end{aligned}$$

Taking the same choices of $\boldsymbol{\phi}$, u and η as in Theorem 3.1, we obtain the final result. $\square$

3.B.4 Proof of Theorem 3.3

To prove Theorem 3.3, we first introduce the following lemma. Recall that $\mathcal{C}_d$ denotes the class of proper, convex lower-semicontinuous functions $\varphi : \mathbb{R}^d \to (-\infty, \infty]$ that are coercive in the sense that $\varphi(\boldsymbol{x}) \to \infty$ as $\|\boldsymbol{x}\| \to \infty$. Recall further from [78, Theorem 2.2] that if P is a distribution on $\mathbb{R}^d$ with $\int_{\mathbb{R}^d} \|\boldsymbol{x}\| \, dP(\boldsymbol{x}) < \infty$ and $P(H) < 1$ for all hyperplanes H, then the strictly convex function $\Gamma : \mathcal{C}_d \to (-\infty, \infty]$ given by

$$\Gamma(\varphi) := \int_{\mathbb{R}^d} \varphi(\boldsymbol{x}) \, dP(\boldsymbol{x}) + \int_{\mathbb{R}^d} e^{-\varphi(\boldsymbol{x})} \, d\boldsymbol{x} \tag{3.50}$$

has a unique minimizer $\varphi^* \in \mathcal{C}_d$ satisfying $\Gamma(\varphi^*) \in \mathbb{R}$.

Lemma 3.2. *Let P be a distribution on $\mathbb{R}^d$ with $\int_{\mathbb{R}^d} \|x\| \, dP(x) < \infty$ and $P(H) < 1$ for all hyperplanes H, and let $\varphi^* := \operatorname{argmin}_{\varphi \in \mathcal{C}_d} \Gamma(\varphi)$. Then*
(1) *For any $\lambda \in [0, 1]$, and $\varphi, \tilde{\varphi} \in \mathcal{C}_d$, we have*

$$\Gamma\big(\lambda\varphi + (1-\lambda)\tilde{\varphi}\big) \le \lambda\Gamma(\varphi) + (1-\lambda)\Gamma(\tilde{\varphi}) - \frac{\lambda(1-\lambda)}{2} \int_{\mathbb{R}^d} e^{-\max\{\varphi(\boldsymbol{x}), \tilde{\varphi}(\boldsymbol{x})\}} \big\{\varphi(\boldsymbol{x}) - \tilde{\varphi}(\boldsymbol{x})\big\}^2 \, d\boldsymbol{x}. \tag{3.51}$$

Here, when $\max\big\{\varphi(\boldsymbol{x}), \tilde{\varphi}(\boldsymbol{x})\big\} = \infty$, we define the integrand to be zero.
(2) *Furthermore, if $\varphi \in \mathcal{C}_d$ is such that $\max\{\varphi(\boldsymbol{x}), \varphi^*(\boldsymbol{x})\} \le \phi^0$ for all $\boldsymbol{x} \in \operatorname{dom}\varphi \cap \operatorname{dom}\varphi^*$, then*

$$\Gamma(\varphi) - \Gamma(\varphi^*) \ge \frac{1}{2} e^{-\phi^0} \int_{\operatorname{dom}\varphi \cap \operatorname{dom}\varphi^*} \big\{\varphi(\boldsymbol{x}) - \varphi^*(\boldsymbol{x})\big\}^2 \, d\boldsymbol{x}. \tag{3.52}$$

Proof. **(1)** Fix $\varphi, \tilde{\varphi} \in \mathcal{C}_d$ with $\max\big\{\Gamma(\varphi), \Gamma(\tilde{\varphi})\big\} < \infty$ (because otherwise the result is clear). For any $M \in \mathbb{R}$, the function $y \mapsto e^{-y}$ is e^{-M}-strongly convex on $y \le M$.

Therefore, for any $\lambda \in [0,1]$, we have $\Gamma\big(\lambda\varphi + (1-\lambda)\tilde{\varphi}\big) \geq \Gamma(\varphi^*) > -\infty$, so

$$\begin{aligned}\Gamma\big(\lambda\varphi + (1-\lambda)\tilde{\varphi}\big) - \lambda\Gamma(\varphi) - (1-\lambda)\Gamma(\tilde{\varphi}) \\ &= \int_{\mathbb{R}^d} \big[e^{-\{\lambda\varphi(\boldsymbol{x}) + (1-\lambda)\tilde{\varphi}(\boldsymbol{x})\}} - \lambda e^{-\varphi(\boldsymbol{x})} - (1-\lambda)e^{-\tilde{\varphi}(\boldsymbol{x})}\big]\, d\boldsymbol{x} \\ &\leq -\frac{\lambda(1-\lambda)}{2} \int_{\mathbb{R}^d} e^{-\max\{\varphi(\boldsymbol{x}),\tilde{\varphi}(\boldsymbol{x})\}} \big\{\varphi(\boldsymbol{x}) - \varphi^*(\boldsymbol{x})\big\}^2\, d\boldsymbol{x},\end{aligned} \tag{3.53}$$

as required.

(2) By (3.53), we have for any $\lambda \in (0,1)$ that

$$\begin{aligned}\frac{\lambda(1-\lambda)}{2} e^{-\phi^0} \int_{\operatorname{dom}\varphi \cap \operatorname{dom}\varphi^*} \big\{\varphi(\boldsymbol{x}) - \varphi^*(\boldsymbol{x})\big\}^2\, d\boldsymbol{x} \\ &\leq \lambda\Gamma(\varphi) + (1-\lambda)\Gamma(\varphi^*) - \Gamma\big(\lambda\varphi + (1-\lambda)\varphi^*\big) \\ &\leq \lambda\big\{\Gamma(\varphi) - \Gamma(\varphi^*)\big\},\end{aligned}$$

where the last inequality follows by definition of φ^*. We deduce that

$$\frac{1}{2} e^{-\phi^0} \int_{\operatorname{dom}\varphi \cap \operatorname{dom}\varphi^*} \big\{\varphi(\boldsymbol{x}) - \varphi^*(\boldsymbol{x})\big\}^2\, d\boldsymbol{x} \leq \frac{\Gamma(\varphi) - \Gamma(\varphi^*)}{1-\lambda}.$$

The result follows on taking $\lambda \searrow 0$. □

Proof of Theorem 3.3. In Lemma 3.2, let P be the empirical distribution of $\{\boldsymbol{x}_i\}_{i=1}^n$. From the proof of Theorem 2 of [61] (see also the proof of Theorem 3.4), Problem (3.50) is equivalent to (3.5) in the sense that $\Gamma(\operatorname{cef}[\boldsymbol{\phi}]) = f(\boldsymbol{\phi})$ for all $\boldsymbol{\phi} \in \mathbb{R}^n$, and $\varphi^* = \operatorname{cef}[\boldsymbol{\phi}^*]$. Now fix $\boldsymbol{\phi} \in \boldsymbol{\Phi}$ and let $\varphi = \operatorname{cef}[\boldsymbol{\phi}]$, so that $\varphi(\boldsymbol{x}) \leq \phi^0$ for all $\boldsymbol{x} \in C_n$. From (3.52), we obtain

$$\frac{1}{2} e^{-\phi^0} \int_{C_n} \big\{\operatorname{cef}[\boldsymbol{\phi}](\boldsymbol{x}) - \operatorname{cef}[\boldsymbol{\phi}^*](\boldsymbol{x})\big\}^2\, d\boldsymbol{x} \leq \Gamma(\operatorname{cef}[\boldsymbol{\phi}]) - \Gamma(\operatorname{cef}[\boldsymbol{\phi}^*]) = f(\boldsymbol{\phi}) - f(\boldsymbol{\phi}^*),$$

as required. □

3.B.5 Proofs of Theorem 3.4 and Theorem 3.5

The proof of Theorem 3.4 is based on following proposition:

Proposition 3.7. *Given $s \in \mathbb{R}$, $\varphi \in \mathcal{C}_d$ and $c > 0$, there exists $\tilde{\varphi} \in \mathcal{C}_d$, such that $\psi_s \circ \varphi(\cdot) = c \cdot \psi_s \circ \tilde{\varphi}(\cdot)$.*

Proof. Given s, φ and c, define

$$\tilde{\varphi}(\cdot) := \begin{cases} \varphi(\cdot) + \log c & \text{if } s = 0 \\ c^{-s}\varphi(\cdot) & \text{if } s \neq 0. \end{cases}$$

Then $\tilde{\varphi} \in \mathcal{C}_d$, and for $\boldsymbol{x} \in \mathcal{D}_s$,

$$\psi_s \circ \tilde{\varphi}(\boldsymbol{x}) = \begin{cases} \big(c^{-s}\varphi(\boldsymbol{x})\big)^{1/s} & \text{if } s < 0 \\ \exp\big(-\varphi(\boldsymbol{x}) - \log c\big) & \text{if } s = 0 \\ \big(-c^{-s}\varphi(\boldsymbol{x})\big)^{1/s} & \text{if } s > 0 \end{cases} = c^{-1} \cdot \psi_s \circ \varphi(\boldsymbol{x}),$$

as required. □

Proof of Theorem 3.4. The proof is split into four steps. The first three steps hold for any $s \in \mathbb{R}$, while in Step 4, we show the convexity of the objective when $s \in [0, 1]$.

Step 1: We claim that any solution $\hat{p}_n$ to (3.21) is supported on C_n, so that $\varphi^*(\boldsymbol{x}) = \infty$ when $\boldsymbol{x} \notin C_n$, where φ^* is the solution to (3.22). Indeed, suppose for a contradiction that $p = \psi_s \circ \varphi \in \mathcal{P}_s(\mathbb{R}^d)$ is such that $\sum_{i=1}^n \log p(\boldsymbol{x}_i) > -\infty$, and that $\int_{C_n} p(\boldsymbol{x})\, d\boldsymbol{x} = c < 1 = \int_{\mathbb{R}^d} p(\boldsymbol{x})\, d\boldsymbol{x}$. We may assume that $c > 0$, because otherwise $p(\boldsymbol{x}) = 0$ for almost all $\boldsymbol{x} \in C_n$, which would mean that $\sum_{i=1}^n \log p(\boldsymbol{x}_i) = -\infty$ since $p \in \mathcal{P}_s(\mathbb{R}^d)$. Define $\bar{\varphi} \in \mathcal{C}_d$ by

$$\bar{\varphi}(\boldsymbol{x}) := \begin{cases} \varphi(\boldsymbol{x}) & \text{if } \boldsymbol{x} \in C_n \\ \infty & \text{otherwise,} \end{cases}$$

so that $\int_{\mathbb{R}^d} \psi_s \circ \bar{\varphi}(\boldsymbol{x})\, d\boldsymbol{x} = \int_{\mathbb{R}^d} \psi_s \circ \varphi(\boldsymbol{x})\, d\boldsymbol{x} = c < 1$. Applying Proposition 3.7 to $\bar{\varphi} \in \mathcal{C}_d$ and $c > 0$, we can find $\tilde{\varphi} \in \mathcal{C}_d$ with $\int_{\mathbb{R}^d} \psi_s \circ \tilde{\varphi}(\boldsymbol{x})\, d\boldsymbol{x} = \int_{\mathbb{R}^d} c^{-1} \cdot \psi_s \circ \bar{\varphi}(\boldsymbol{x})\, d\boldsymbol{x} = 1$,

and

$$\sum_{i=1}^n \log \psi_s \circ \tilde{\varphi}(\boldsymbol{x}_i) = \sum_{i=1}^n \log \psi_s \circ \varphi(\boldsymbol{x}_i) - n \log c > \sum_{i=1}^n \log \psi_s \circ \varphi(\boldsymbol{x}_i).$$

This establishes our desired contradiction.

Step 2: We claim that any solution φ^* to (3.22) satisfies

$$\varphi^* = \operatorname*{argmin}_{\substack{\varphi \in \mathcal{C}_d : \operatorname{Im}(\varphi) \subseteq \mathcal{D}_s \cup \{\infty\}, \\ \operatorname{dom} \varphi = C_n}} \left\{ -\frac{1}{n} \sum_{i=1}^n \log \psi_s \circ \varphi(\boldsymbol{x}_i) + \int_{C_n} \psi_s \circ \varphi(\boldsymbol{x}) \, d\boldsymbol{x} \right\}. \tag{3.54}$$

Indeed, for any $\varphi \in \mathcal{C}_d$ such that $\operatorname{dom} \varphi = C_n$ and $\int_{C_n} \psi_s \circ \varphi(\boldsymbol{x}) \, d\boldsymbol{x} = c \neq 1$, we can again apply Proposition 3.7 to φ and c to obtain $\tilde{\varphi}$. Then

$$\begin{aligned} -\frac{1}{n} \sum_{i=1}^n \log \psi_s \circ \varphi(\boldsymbol{x}_i) + \int_{C_n} \psi_s \circ \varphi(\boldsymbol{x}) \, d\boldsymbol{x} &= -\frac{1}{n} \sum_{i=1}^n \log \psi_s \circ \varphi(\boldsymbol{x}_i) + c \\ > -\frac{1}{n} \sum_{i=1}^n \log \psi_s \circ \varphi(\boldsymbol{x}_i) + \log c + 1 &= -\frac{1}{n} \sum_{i=1}^n \log \psi_s \circ \tilde{\varphi}(\boldsymbol{x}_i) + \int_{C_n} \psi_s \circ \tilde{\varphi}(\boldsymbol{x}) \, d\boldsymbol{x}, \end{aligned}$$

so $\int_{C_n} \psi_s \circ \varphi^*(\boldsymbol{x}) \, d\boldsymbol{x} = 1$, which establishes our claim.

Step 3: Letting $\boldsymbol{\phi}^* = (\phi_1^*, \ldots, \phi_n^*)$ denote an optimal solution to (3.23), we claim that $\operatorname{cef}[\boldsymbol{\phi}^*](\boldsymbol{x}_i) = \phi_i^*$ holds for all $i \in [n]$. Indeed, for any $\boldsymbol{\phi} = (\phi_1, \ldots, \phi_n) \in \mathcal{D}_s^n$, if there exists $i^* \in [n]$ such that $\operatorname{cef}[\boldsymbol{\phi}](\boldsymbol{x}_{i^*}) \neq \phi_{i^*}$, then we can define $\tilde{\boldsymbol{\phi}} = (\tilde{\phi}_1, \ldots, \tilde{\phi}_n) \in \mathcal{D}_s^n$ such that $\tilde{\phi}_i = \operatorname{cef}[\boldsymbol{\phi}](\boldsymbol{x}_i)$ for all $i \in [n]$. We now claim that $\operatorname{cef}[\tilde{\boldsymbol{\phi}}] = \operatorname{cef}[\boldsymbol{\phi}]$. On the one hand, by (3.3), $\tilde{\phi}_i = \operatorname{cef}[\boldsymbol{\phi}](\boldsymbol{x}_i) \leq \phi_i$ for any $i \in [n]$. From the LP expression (Q_0), $\operatorname{cef}[\tilde{\boldsymbol{\phi}}] \leq \operatorname{cef}[\boldsymbol{\phi}]$. On the other hand, since $\operatorname{cef}[\boldsymbol{\phi}](\cdot)$ is a convex function with $\operatorname{cef}[\boldsymbol{\phi}](\boldsymbol{x}_i) = \tilde{\phi}_i \leq \tilde{\phi}_i$ for any i, we have $\operatorname{cef}[\tilde{\boldsymbol{\phi}}](\cdot) \geq \operatorname{cef}[\boldsymbol{\phi}](\cdot)$. It follows that $\operatorname{cef}[\tilde{\boldsymbol{\phi}}] = \operatorname{cef}[\boldsymbol{\phi}]$, and $\tilde{\boldsymbol{\phi}}$ with a smaller objective than $\boldsymbol{\phi}$. This establishes our claim, and shows that (3.23) is equivalent to (3.54) in the sense that $\hat{p}_n$ and $\boldsymbol{\phi}^*$ satisfy $\hat{p}_n = \psi_s(\operatorname{cef}[\boldsymbol{\phi}^*])$.

Step 4: When $s = 0$, the function $\boldsymbol{\phi} \mapsto \frac{1}{n} \mathbf{1}^\top \boldsymbol{\phi}$ is convex on $\mathbb{R}^n$; when $s > 0$, the function $\boldsymbol{\phi} \mapsto -\frac{1}{ns} \sum_{i=1}^n \log(-\phi_i)$ is convex on $(-\infty, 0]^n$. Moreover, when $s \leq 1$, the function ψ_s is decreasing and convex, and since $\boldsymbol{\phi} \mapsto \operatorname{cef}[\boldsymbol{\phi}](\boldsymbol{x})$ is concave for every $\boldsymbol{x} \in \mathbb{R}^n$, the result follows. $\square$

3.C Background on shape-constrained inference

Entry points to the field of nonparametric inference under shape constraints include the book by [100], as well as the 2018 special issue of the journal *Statistical Science* [196]. Other canonical problems in shape constraints that involve non-trivial computational issues include isotonic regression ([41, 227, 49, 81, 21, 224, 108, 181]) and convex regression ([121, 200, 44, 102, 110, 84, 52]), or combinations and variants of these ([55]).

Beyond papers already discussed, early theoretical work on log-concave density estimation includes [180], [76], [215], [60], [78], [199], [197] and [54]. Sometimes, the class $\mathcal{P}_d$ is considered as a special case of the class of s-concave densities ([137, 201, 109, 73, 107]); see also Section 3.5. Much recent work has focused on rates of convergence, which are best understood in the Hellinger distance d_{H}, given by

$$d_{\mathrm{H}}^2(p,q) := \int_{\mathbb{R}^d} (p^{1/2} - q^{1/2})^2.$$

For the case of correct model specification, i.e. where $\hat{p}_n$ is computed from an independent and identically distributed sample of size n from $p_0 \in \mathcal{P}_d$, it is now known [134, 138] that

$$\sup_{p_0 \in \mathcal{P}_d} \mathbb{E} d_{\mathrm{H}}^2(\hat{p}_n, p_0) \leq K_d \cdot \begin{cases} n^{-4/5} & \text{when } d = 1 \\ n^{-2/(d+1)} \log n & \text{when } d \geq 2, \end{cases}$$

where $K_d > 0$ depends only on d, and that this risk bound is minimax optimal (up to the logarithmic factor when $d \geq 2$). See also [47] for an earlier result in the case $d \geq 4$, and [223] for an alternative approach to high-dimensional log-concave density estimation that seeks to evade the curse of dimensionality in the additional presence of symmetry constraints. It is further known that when $d \leq 3$, the log-concave maximum likelihood estimator can adapt to certain subclasses of log-concave densities, including log-concave densities whose logarithms are piecewise affine [133, 85]. See also [16] for recent work on extensions to the misspecified setting (where the true distribution

from which the data are drawn does not have a log-concave density).

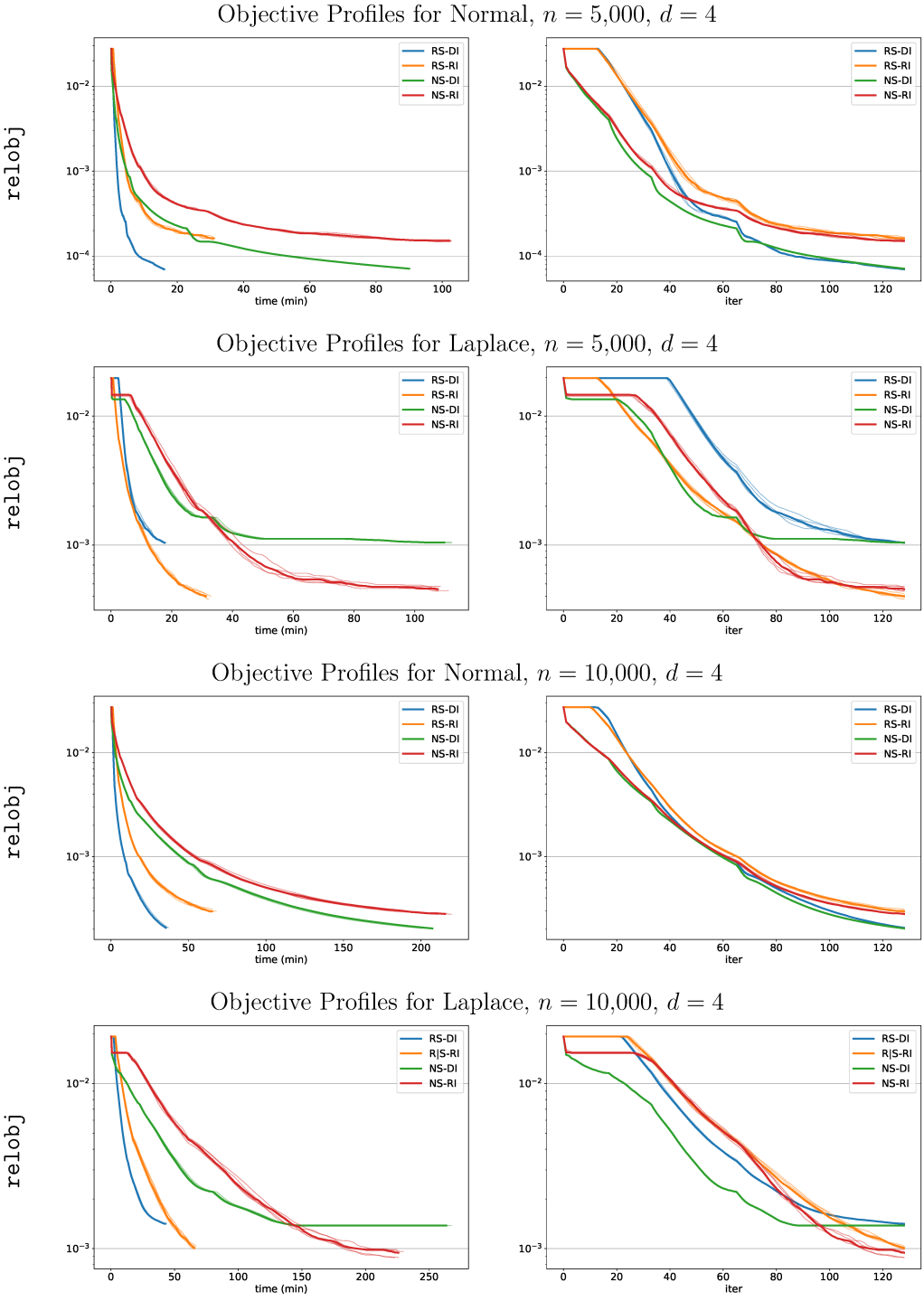

Figure 3-2: Plots on a log-scale of Relative Objective versus time (mins) [left panel] and number of iterations [right panel]. For each of our four synthetic data sets, we ran five repetitions of each algorithm, so each bold line corresponds to the median of the profiles of the corresponding algorithm, and each thin line corresponds to the profile of one repetition. For the right panel, we show the profiles up to 128 iterations.

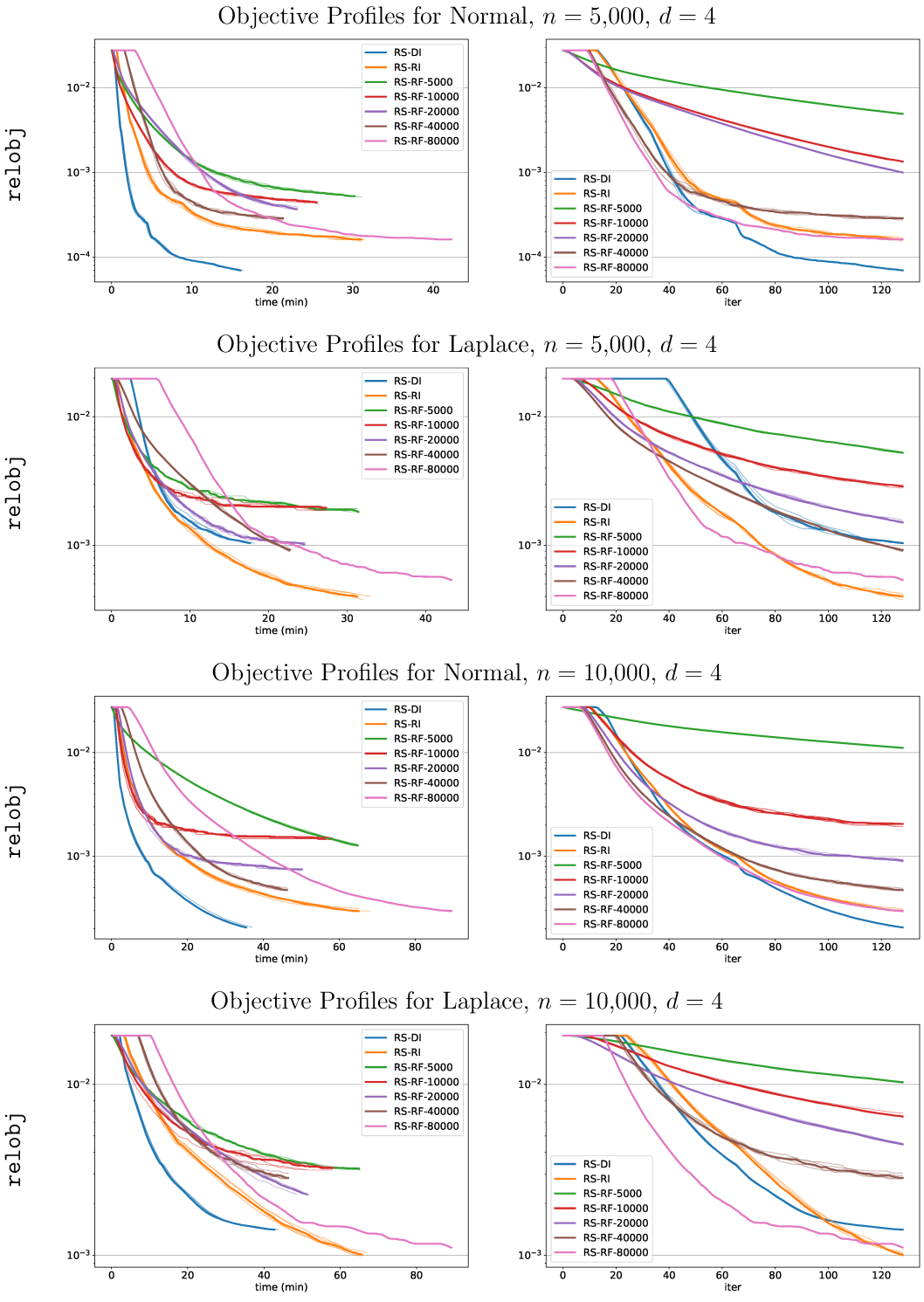

Figure 3-3: Plots on a log-scale of Relative Objective versus time (mins) [left panel] and number of iterations [right panel].

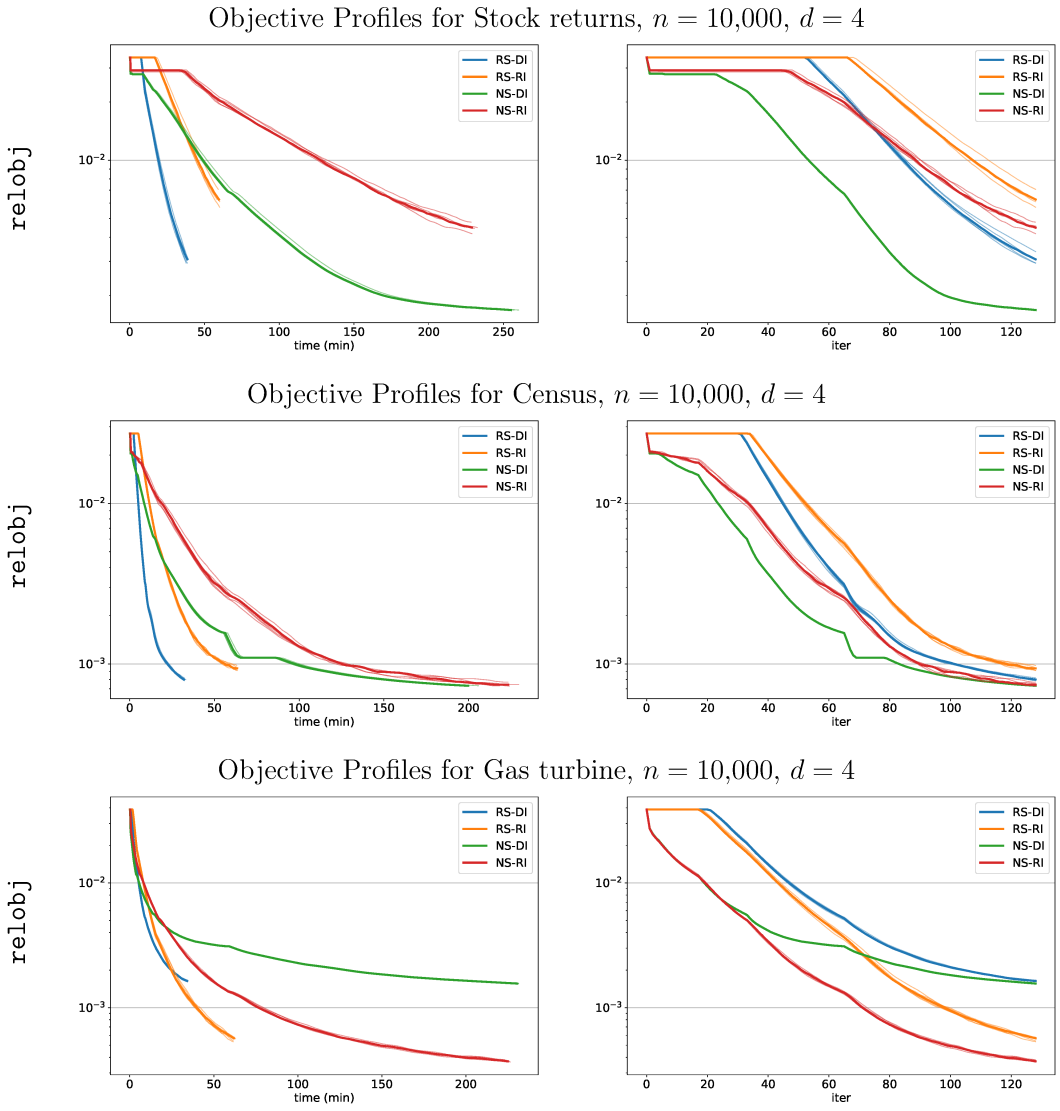

Figure 3-4: Additional plots on a log-scale of Relative Objective versus time (mins) [left panel] and number of iterations [right panel]. Details are given in the caption of Figure 3-2.

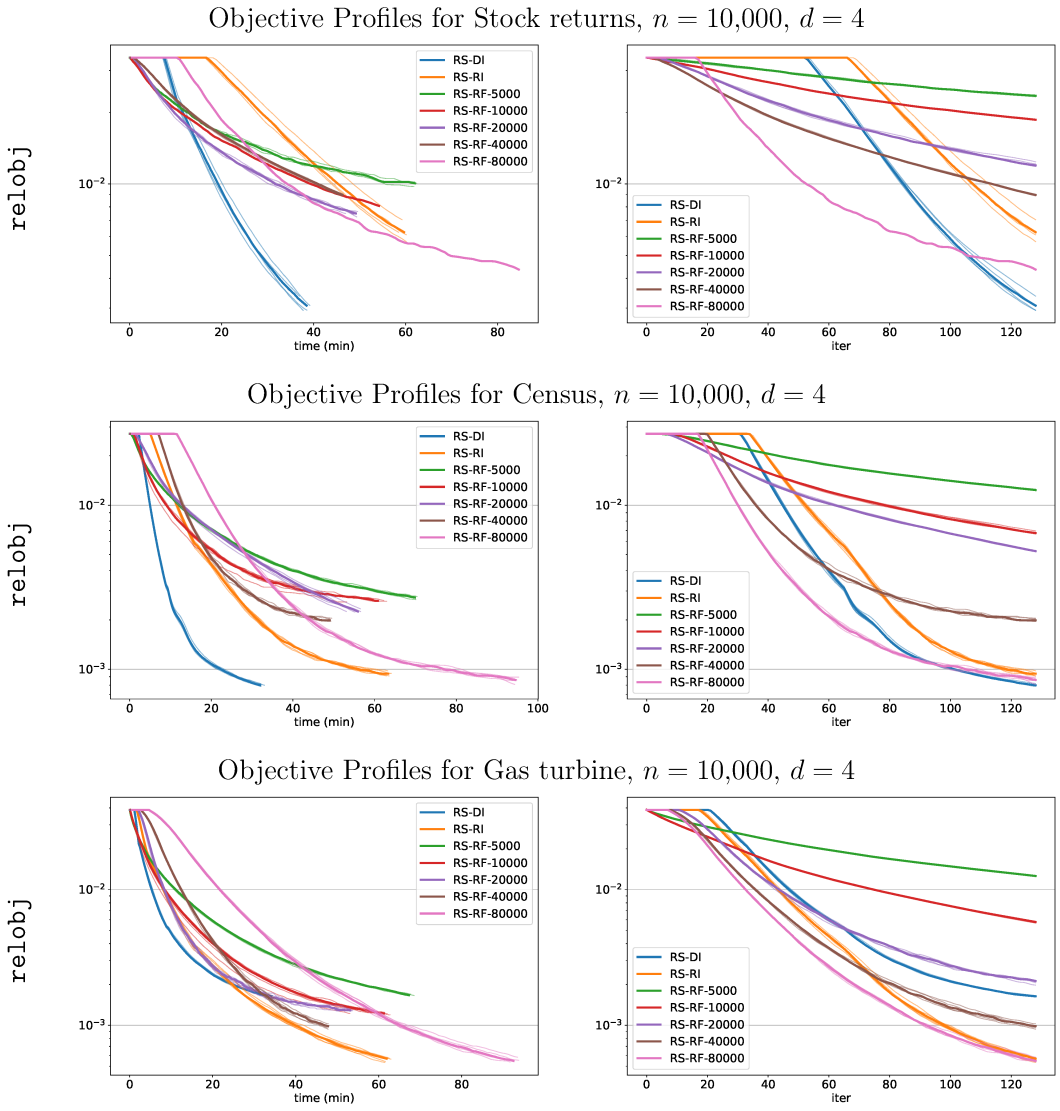

Figure 3-5: Plots on a log-scale of Relative Objective versus time (mins) [left panel] and number of iterations [right panel].

Chapter 4

Guassian Graphical Models: A Scalable Framework Based on Combinatorial Optimization

This is a joint work with Kayhan Behdin and Rahul Mazumder.

4.1 Introduction

Gaussian Graphical Models (GGM), due to Dempster [68], are amongst the most widely used tools to analyze continuous multivariate random systems ([93, Chapter 17] and [213, Chapter 11]). Formally, in a GGM, we are given n data points $\boldsymbol{x}^{(1)}, \cdots, \boldsymbol{x}^{(n)} \in \mathbb{R}^p$ from a multivariate normal distribution as

$$\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(n)} \overset{\text{iid}}{\sim} \mathcal{N}(\mathbf{0}, (\boldsymbol{\Theta}^*)^{-1}), \quad \boldsymbol{\Theta}^* \in \mathbb{S}_+^p \tag{4.1}$$

where $\mathbb{S}_+^p$ is the set of $p \times p$ positive definite matrices. The goal in the GGM problem is to estimate the matrix $\boldsymbol{\Theta}^*$, known as the precision matrix, as this matrix contains the information required to recover the normal distribution generating the data.

From both interpretability and statistical perspectives, a sparse estimation of $\boldsymbol{\Theta}^*$ (i.e., one with only a few nonzero coordinates) is more desirable [68] — through-

out this chapter, we assume $\mathbf{\Theta}^*$ is sparse. A zero entry in Θ^* indicates conditional independence: For a pair (i, j), $\theta^*_{ij} = 0$ if and only if features i, j are independent conditioned on the other variables. Our goal is to provide a precision matrix $\mathbf{\Theta}^*$ such that it is sparse (i.e., has few nonzeros). The problem of sparse precision matrix estimation has garnered significant attention in the statistics and machine learning literature [90, 183, 83]. Performance of an estimation $\hat{\mathbf{\Theta}}$ of $\mathbf{\Theta}^*$ can be measured by different criteria. A common metric is the Frobenius norm estimation error defined as $\|\mathbf{\Theta}^* - \hat{\mathbf{\Theta}}\|_F^2$. We also seek to estimate the location of nonzeros of $\mathbf{\Theta}^*$ (i.e., the support).

In what follows, we first present an overview of current algorithms for sparse GGMs and then summarize our key contributions in this chapter.

4.1.1 Background and Literature Review

Numerous algorithms have been proposed for GGMs. Generally, these methods aim to minimize a penalized data fidelity loss, where the penalty encourages sparsity in the solution. One of the most popular approaches to GGMs is ℓ_1-regularized likelihood maximization, known as Graphical Lasso [90]. Graphical Lasso is known to enjoy good statistical and computational properties [189, 159]. Another approach is constrained ℓ_1-norm minimization [45], known as CLIME. CLIME estimator can be calculated by solving a series of linear programs, and overall the CLIME estimator leads to good statistical guarantees. Another approach to GGM is the node-wise linear regression framework of [163] which requires solving p-many sparse linear regression problems (i.e. Lasso).

In this chapter, we focus on an interesting but less-understood approach to GGMs, the pseudo-likelihood approach. The notion of pseudo-likelihood was introduced by Besag [35] in the context of spatial analysis. In the pseudo-likelihood analysis, instead of directly working with the likelihood of the data, the likelihood function is approximated by the multiplication of conditional likelihood functions of each variable, given the rest. Pseudo-likelihood analysis with ℓ_1 regularization was first applied to GGMs by [183]. [183] presents an asymptotic analysis of their algorithm when $n \to \infty$. Oth-

ers have proposed algorithms based on pseudo-likelihood, for example, see [91, 132]. However, the statistical properties of these methods are less understood compared to more common methods.

A recent approach to GGMs is based on discrete optimization. Since the work of [28], there has been a renewed interest in exploring statistical problems that admit combinatorial structure using tools from Mixed Integer Programming (MIP)[220]. In particular, MIP techniques have proved useful in sparse learning problems, where the sparsity structure can be encoded by binary variables. Sparse linear regression [32, 119] and sparse principal component analysis [20] are among examples where MIP methods have been successful. However, there has been little exploration regarding MIP formulations for GGMs. [29] consider an ℓ_0-constrained maximum likelihood approach. They propose an MIP algorithm for their formulation, however, their algorithm only scales to problems with $p \approx 100$.

In terms of statistical performance, to have a consistent estimation in terms of Frobenius norm, we need $n \gtrsim kp \log p$ samples [194], where k is the number of nonzeros in each row of $\mathbf{\Theta}^*$. This implies that a consistent estimation is only possible when $kp/n \to 0$ which corresponds to the classical low-dimensional setting. In the high-dimensional setting, estimating the correct support is more interesting. In general, under certain non-degeneracy conditions, $n \gtrsim k \log p$ samples are required for a consistent estimation of the support of $\mathbf{\Theta}^*$ [218]. This shows that support recovery is possible even when $p/n \to \infty$.

4.1.2 Outline of the Approach and Contributions

Despite the promising results of pseudo-likelihood estimators, the theoretical and algorithmic understanding of such estimators has remained limited. In this chapter, we propose a new pseudo-likelihood-based estimator that enjoys both good statistical guarantees and computational performance. To this end, in a departure from current literature, we consider an ℓ_0 regularized version of the pseudo-likelihood function. We show that this estimator can be reduced to an MIP, where a convex objective function is minimized over a Mixed Integer Second-Order Conic (MISOC) constraint

set. We then study the statistical properties of the estimator and develop a scalable algorithm for the MIP.

We analyze our estimator from estimation and variable selection points of view. In terms of estimation, our estimator achieves Frobenius error bound scaling as $kp \log p/n$. As for the variable selection performance, we show that under certain regularity conditions, if $n \gtrsim k \log p$, our estimator is able to recover the support of $\mathbf{\Theta}^*$ correctly with high probability. An interesting property of our estimator is that the non-degeneracy condition needed for consistent variable selection for our method is milder compared to the ones appearing in the literature. This is due to certain symmetry structures that we enforce on the solution.

To solve the MIP for the estimator, we develop a specialized Branch-and-Bound (BnB) solver that does not rely on commercial MIP solvers. Following [119], our BnB framework solves a version of our MIP based on a perspective reformulation [101] of the ℓ_0-regularized pseudo-likelihood function. We propose a coordinate-descent (CD) algorithm to solve the node relaxations as well as obtaining a fast approximate integral solution. With both logarithmic term and quadratic-over-linear structure in the objective (see Section 4.2), the existing convergence theory of CD algorithm is not directly applicable. We provide the computational guarantee of CD algorithm for node relaxations, which also applies to a convex reformulation of the estimator proposed in [91]. We also make use of active set updates to reduce the complexity of CD algorithm by exploiting the structured sparsity in the statistical problem, as well as shared information across the BnB tree. Furthermore, we propose a novel method to efficiently generate dual bounds in the BnB tree from primal solutions. Our dual method handles extra symmetry constraint and special terms in the objective function arsing from the statistical structure, compared to the dual bounds derived for the sparse regression in [119].

Finally, we perform numerical experiments on both synthetic and real datasets, and compare results with other existing methods in terms of both statistical performance and computational efficiency. The results indicate that our optimization framework is scalable to problems with $p \approx 10{,}000$ while it is faster or compara-

ble to polynomial-time methods. Moreover, our proposed estimator provides better statistical performance on synthetic and real datasets.

Our contributions in this chapter can be summarized as follows:

1. We propose a new estimator for GGMs as an ℓ_0-regularized pseudo-likelihood problem. We show our estimator can be written as an MIP.
2. We discuss statistical properties of our estimator and discuss how our new estimator can improve upon existing algorithms in terms of estimation and variable selection.
3. We develop and implement an optimization framework for our estimator, including heuristic solvers and a specialized nonlinear branch-and-bound method. Our framework is open-source and does not use any commercial solver.
4. Our numerical experiments show that the proposed estimator outperforms existing (polynomial-time) methods for GGMs in terms of runtime and statistical performance.

Organization of chapter In Section 4.2, we introduce our proposed estimator, and reformulate it into a MIP problem based on perspective reformulation. In Section 4.3, we provide an efficient computational framework for our proposed estimator. In Section 4.4, we analyze the statistical properties of our proposed estimator. In Section 4.5, we present various numerical experiments on both synthetic and real datasets, showing the benefits of our estimator from both statistical and computational perspectives. The derivations and proofs in the computational and statistical parts are deferred to Appendix 4.A and 4.B.

Notations For $\boldsymbol{A} \in \mathbb{R}^{p_1 \times p_2}$ and $S_1 \subseteq [p_1], S_2 \subseteq [p_2]$, denote by $\boldsymbol{A}_{S_1,S_2}$ the submatrix of $\boldsymbol{A}$ with rows sampled in S_1 and columns sampled in S_2. $\mathcal{B}(p)$ denotes the unit Euclidean ball of dimension p. Let $\mathbb{S}^p$ denote the set of symmetric matrices in $\mathbb{R}^{p\times p}$. We let $\mathbf{1}\{x \neq 0\}$ denote the indicator function, i.e. $\mathbf{1}\{x \neq 0\} = 1$ if $x \neq 0$; otherwise, $\mathbf{1}\{x \neq= 0\} = 0$. We let $\chi\{a \in A\}$ denote the characteristic function, i.e. $\chi\{a \in A\} = 0$ if $a \in A$; otherwise, $\chi\{a \in A\} = \infty$.

4.2 Proposed Estimator

Our formulation of the GGM problem is based on the idea of pseudo-likelihood. If $\boldsymbol{X} \in \mathbb{R}^{n\times p}$ has independent rows of $\mathcal{N}(\mathbf{0}, (\boldsymbol{\Theta}^*)^{-1})$ for some $\boldsymbol{\Theta}^* \in \mathbb{S}^p_+$, the conditional distribution of each variable, given the rest, follow the normal distribution

$$\boldsymbol{x}_j | \{\boldsymbol{x}_i\}_{i\neq j} \sim \mathcal{N}\left(\sum_{i\neq j} \beta^*_{ij}\boldsymbol{x}_i, (\sigma^*_j)^2 \boldsymbol{I}_n\right) \tag{4.2}$$

where

$$\beta^*_{ij} = -\frac{\theta^*_{ji}}{\theta^*_{jj}} \quad i \neq j \in [p], \quad (\sigma^*_j)^2 = \frac{1}{\theta^*_{jj}} \quad j \in [p]. \tag{4.3}$$

We note that $\beta^*_{ij} \neq 0$ if and only if $\theta^*_{ji} \neq 0$ and as $\boldsymbol{\Theta}^*$ is sparse, most values of β^*_{ij} are zero. As a result, we propose an ℓ_0-constrained estimator:

$$\min_{\beta_{ij},\sigma_j} \quad \sum_{j=1}^{p}\left[\log(\sigma_j) + \frac{1}{n}\frac{1}{2\sigma_j^2}\left\|\boldsymbol{x}_j - \sum_{i:i\neq j}\beta_{ij}\boldsymbol{x}_i\right\|_2^2\right] \tag{4.4a}$$

$$\text{s.t.} \quad \beta_{ij}\sigma_i^2 = \beta_{ji}\sigma_j^2, \quad \beta_{ii} = 0, \quad i \neq j \tag{4.4b}$$

$$|\{i : i \neq j, \beta_{ij} \neq 0\}| \leq k, \quad j \in [p]. \tag{4.4c}$$

In Problem (4.4), the objective function is the summation of negative log-likelihood functions for Normal distributions given in (4.2). Constraint (4.4b) enforces the symmetric structure based on the fact $\beta^*_{ij}(\sigma^*_i)^2 = \beta^*_{ji}(\sigma^*_j)^2$. Finally, constraint (4.4c) enforces that β and consequently θ are sparse, as $\beta^*_{ij} \neq 0 \Leftrightarrow \theta^*_{ji} \neq 0$. We investigate statistical properties of this estimator in Section 4.4. In what follows, we present a convex mixed integer formulation of the estimator introduced in (4.4).

4.2.1 A convex mixed integer reformulation

Problem (4.4) in its current form has a non-convex objective function and involves nonlinear symmetry constraints. However, this issue can be remedied by simple change of variables [145, 132] — $\theta_{jj} = 1/\sigma_j^2$ and $\beta_{ij} = -\theta_{ji}/\theta_{jj}$. Under this mapping, the symmetry constraint (4.4b) simplifies to the matrix $\boldsymbol{\Theta}$ being symmetric.

Moreover, instead of the sparsity constraint (4.4c), we impose an $\ell_0 - \ell_2$ penalty [160] on the off-diagonals of $\boldsymbol{\Theta}$ to enforce the sparsity. Such penalty is used to help prevent over-fitting in the regime of low-signal-to-noise ratios in the context of sparse regression. As we discuss throughout the chapter, this regularization leads to improved computational properties. Overall, our reformulation of Problem (4.4) is given as:

$$\min_{\boldsymbol{\Theta} \in \mathbb{S}^p} \; F_0(\boldsymbol{\Theta}) = \sum_{i=1}^{p} \big(-\log(\theta_{ii}) + \frac{1}{\theta_{ii}} \|\tilde{\boldsymbol{X}} \boldsymbol{\theta}_i\|^2 \big) + \sum_{i<j} \big(\lambda_0 \mathbf{1}\{\theta_{ij} \neq 0\} + \lambda_2 \theta_{ij}^2 \big) \tag{4.5}$$

where $\tilde{\boldsymbol{X}} = \frac{1}{\sqrt{n}} \boldsymbol{X}$ and $\lambda_0, \lambda_2 \geq 0$ are regularization coefficients that should be selected. Next, we introduce a perspective reformulation of Problem (4.5). Perspective formulations [88, 4, 101] are helpful in terms of stronger MIP relaxations, and they have been used recently in a specialized BnB framework for sparse regression [119]. To this end, we introduce the auxiliary binary variables z_{ij} that encode the sparsity and consider the following perspective reformulation of Problem (4.5):

$$\begin{aligned}
\min_{\boldsymbol{\Theta}, \boldsymbol{z}, \boldsymbol{s}} \; & F_{\mathsf{mio}}(\boldsymbol{\Theta}, \boldsymbol{z}, \boldsymbol{s}) = \sum_{i=1}^{p} \big(-\log(\theta_{ii}) + \frac{1}{\theta_{ii}} \|\tilde{\boldsymbol{X}} \boldsymbol{\theta}_i\|^2 \big) + \sum_{i<j} \big(\lambda_0 z_{ij} + \lambda_2 s_{ij} \big), \qquad (4.6) \\
\text{s.t.} \; & s_{ij} z_{ij} \geq \theta_{ij}^2, \; |\theta_{ij}| \leq M z_{ij}, \quad \forall j \neq i \\
& \theta_{ij} = \theta_{ji}, \quad \forall i \neq j \\
& z_{ij} \in \{0, 1\}, \; s_{ij} \geq 0, \quad \forall j \neq i.
\end{aligned}$$

Further details and benefits of using such a perspective reformulation along with big-M constraint can be found in [119]. In Section 4.3, we present an optimization framework for Problem (4.6).

4.3 Computational Framework

In this section, we will focus on a specialized scalable branch-and-bound (BnB) framework for efficiently solving Problem (4.6). In Section 4.3.1, we discuss related work on nonlinear BnB and provide an overview of our specialized BnB framework. In Sec-

tion 4.3.2, we study the formulations of node relaxations of Problem (4.6) in the BnB. We then present an efficient coordinate descent algorithm along with the active-set update for both node relaxations and heuristic solver in Section 4.3.3, and provide more details in Sections 4.3.4 and 4.3.6. In Section 4.3.5, we show how to obtain dual bounds from primal solutions to node relaxations.

4.3.1 Related work and overview of BnB framework

Our BnB framework follows the high-level ideas proposed by the prior work of [119]. In their paper, they propose a specialized BnB framework for the sparse regression, which consists of a highly-scalable primal-based CD algorithm along with the active-set update and gradient screening. There are several challenges arising from the differences of the sparse regression and pseudolikelihood function for GGM. Specifically, the extra logarithm term, quadratic-over-linear structure and symmetry constraint lead to complicated CD updates, unknown convergence property of CD algorithm and complicated dual bounds.

Overview of Nonlinear BnB: For self-containedness of this chapter, we provide a brief overview of nonlinear BnB framework. Nonlinear BnB is a general framework for solving mixed integer nonlinear programs [22]. The algorithm starts by solving the root relaxation (4.9) of (4.6). Then, the algorithm chooses a branching variable, say $z_{k\ell}$ and create two new nodes (optimization subproblems): one with $z_{k\ell} = 0$ and the other with $z_{k\ell} = 1$, where all the other z_{ij}'s are relaxed to the interval $[0, 1]$. The algorithm then proceeds recursively: for every unvisited node, it solves the corresponding optimization problem and then branch on a new fractional variable (if any) to create new nodes. This leads to a search tree with nodes corresponding to optimization subproblems and edges representing branching decisions.

While growing the search tree, BnB prunes a node either (a) the relaxation at the current node has an integral $\boldsymbol{z}$ or (b) the objective of the current relaxation exceeds the best available upper bound on (4.6), which can be obtained from any feasible integral solution to the problem.

Our strategies:

- **Node relaxations:** Similar to [119], one can show that the node relaxations of Problem (4.6) can be written in the $\boldsymbol{\Theta}$-space instead of the extended $(\boldsymbol{\Theta}, \boldsymbol{z}, \boldsymbol{s})$-space. The formulations are studied in Section 4.3.2.
- **Convex relaxation solver:** To solve the node relaxations, we develop a scalable coordinate descent (CD) algorithm with active set update. The algorithm exploits and shares warm starts and active set information across the BnB tree to further improve the efficiency. Such active set strategies and warm starts turn out to be keys to the speedup of our approach compared to a generic BnB framework. Our algorithm will be described in Section 4.3.3 and additional computational details and convergence guarantee will be provided in Section 4.3.4.
- **Dual bounds:** Dual bounds of the node relaxation problem provides important lower bound information for search space pruning. We develop a novel method to compute dual bounds from the primal solutions, using the convex conjugate of the regularizer derived in [119]. See Section 4.3.5.
- **Heuristic solver and upper bounds:** Better upper bounds can lead to more aggressive pruning in the search tree, thus reducing the BnB running time. At each node of BnB, we attempt to improve the upper bound based on the solution $\hat{\boldsymbol{\Theta}}$ to the current node's relaxation problem. To be more specific, let $\mathcal{S}$ be some sparsity pattern induced by the current solution $\hat{\boldsymbol{\Theta}}$. Then, we use a heuristic solver to obtain an approximate solution to

$$\begin{aligned} \min_{\boldsymbol{\Theta} \in \mathbb{S}^p} \; & \sum_{i=1}^{p} \big(-\log \theta_{ii} + \frac{1}{\theta_{ii}} \|\tilde{\boldsymbol{X}} \boldsymbol{\theta}_i\|^2 \big) + \sum_{(i,j) \in \mathcal{S}} \big(\lambda_0 \mathbf{1}\{\theta_{ij} \neq 0\} + \lambda_2 \theta_{ij}^2 \big), \\ & \text{s.t. } |\theta_{ij}| \leq M, \; \forall (i,j) \in \mathcal{S}; \quad \theta_{ij} = 0, \; \forall (i,j) \in \mathcal{S}^c, \end{aligned} \tag{4.7}$$

  starting from the current solution $\hat{\boldsymbol{\Theta}}$. We use the same method introduced in Section 4.3.3 to solve Problem (4.7). Moreover, a better initialization might lead to a better upper bound as this problem is not convex. We will discuss more details about initialization in Section 4.3.6.

4.3.2 Formulations in BnB

In this section, we study the convex relaxation of (4.6) at each node of the BnB search tree. We start with the root relaxation, where all the binary variables $\boldsymbol{z}$ are relaxed to the interval $[0,1]$. It can be shown [119] that the root relaxation in terms of the variables $(\boldsymbol{\Theta}, \boldsymbol{z}, \boldsymbol{s})$ can be expressed in the $\boldsymbol{\Theta}$ space instead, using the regularizer ψ:

$$\begin{aligned}\psi(\theta;\lambda_0,\lambda_2,M) := \min_{z,s} \quad & \lambda_0 z + \lambda_2 s \\ \text{s.t.} \quad & sz \geq \theta^2, |\theta| \leq Mz, z \in [0,1] \\ = & \begin{cases} 2\sqrt{\lambda_0\lambda_2}|\theta| & \text{if } |\theta| \leq \sqrt{\lambda_0/\lambda_2} \leq M \\ \lambda_0 + \lambda_2\theta^2 & \text{if } \sqrt{\lambda_0/\lambda_2} \leq |\theta| \leq M \\ (\lambda_0/M + \lambda_2 M)|\theta| & \text{if } |\theta| \leq M \leq \sqrt{\lambda_0/\lambda_2} \\ \infty & \text{if } |\theta| > M. \end{cases}\end{aligned} \tag{4.8}$$

This leads to the root relaxation problem as follows:

$$\min_{\boldsymbol{\Theta}\in\mathbb{S}^p} \; F_{\mathsf{root}}(\boldsymbol{\Theta}) = \sum_{i=1}^{p}\big(-\log(\theta_{ii}) + \frac{1}{\theta_{ii}}\|\tilde{\boldsymbol{X}}\boldsymbol{\theta}_i\|^2\big) + \sum_{i<j}\psi(\theta_{ij};\lambda_0,\lambda_2,M), \tag{4.9}$$

We note that this regularizer is closely related to the reverse Huber penalty [178] (see also [72]).

Node relaxation within the BnB tree: For each node within the BnB tree, the node relaxation is similar to the root relaxation, except that some of $\boldsymbol{z}_{ij}$'s are fixed to 0 and 1. Let $[\underline{z}_{ij}, \bar{z}_{ij}]$ be the range of z_{ij} at each node relaxation[1], the corresponding node relaxation problem is

$$\min_{\boldsymbol{\Theta}\in\mathbb{S}^p} \; F_{\mathsf{node}}(\boldsymbol{\Theta}) = \sum_{i=1}^{p}\big(-\log(\theta_{ii}) + \frac{1}{\theta_{ii}}\|\tilde{\boldsymbol{X}}\boldsymbol{\theta}_i\|^2\big) + \sum_{i<j} g(\theta_{ij};\lambda_0,\lambda_2,M,\underline{z}_{ij},\bar{z}_{ij}), \tag{4.10}$$

[1]For example, if z_{ij} is relaxed to $[0,1]$, then $\underline{z}_{ij} = 0$ and $\bar{z}_{ij} = 1$; if z_{ij} is fixed to 0 or 1, then $\underline{z}_{ij} = \bar{z}_{ij} = 0$ or 1.

where

$$g(\theta;\lambda_0,\lambda_2,M,\underline{z},\bar{z}) = \begin{cases} \psi(\theta;\lambda_0,\lambda_2,M) & \text{if } \underline{z}=0,\bar{z}=1 \\ \varphi(\theta;z,\lambda_0,\lambda_2,M) & \text{if } \underline{z}=\bar{z}=z, \end{cases} \tag{4.11}$$

in which

$$\varphi(\theta;z,\lambda_0,\lambda_2,M) = \begin{cases} \chi\{\theta=0\} & \text{if } z=0 \\ \chi\{|\theta|\le M\}+\lambda_0+\lambda_2\theta^2 & \text{if } z=1. \end{cases} \tag{4.12}$$

To this end, we consider the following unified formulation

$$\min_{\boldsymbol{\Theta}\in\mathbb{S}^p} \; F(\boldsymbol{\Theta}) = \sum_{i=1}^{p} -\log(\theta_{ii}) + \frac{1}{\theta_{ii}}\|\tilde{\boldsymbol{X}}\boldsymbol{\theta}_i\|^2 + \sum_{i<j} h_{ij}(\theta_{ij}), \tag{4.13}$$

where h_{ij} is some regularizer. Problem (4.13) encompasses the original problem (4.5), the root relaxation problem (4.9), the node relaxation problem (4.10) and the problem for incumbent solving (4.7) as special cases. Furthermore, when $h_{ij}(\theta)=\lambda_1|\theta|$, (4.13) is equivalent to the symmetric lasso procedure proposed by [91]; the difference is that in [91], they consider a nonconvex formulation wrt $\sigma^{jj}=\frac{1}{\theta_{ii}}$ and off-diagonals $\tilde{\boldsymbol{\Theta}}$ of $\boldsymbol{\Theta}$.

In next section, we will develop a scalable active-set coordinate descent algorithm for solving or approximately solving (4.13), depending on the convexity of F.

4.3.3 Active-set Coordinate Descent

Due to the separability of the (nonsmooth) regularizers h_{ij}, Problem (4.13) is amenable to the cyclic CD [206] with full minimization in every coordinate in the lower triangular part of $\boldsymbol{\Theta}$. CD-type methods are widely used for solving huge-scale optimization problems in statistical learning, especially those problems with sparsity structure, due to their inexpensive iteration updates and capability of exploiting problem structure. For example, they have been used to solve the Lasso problem [94, 198], the support vector machines [126, 48], and the graphical Lasso [159].

As presented in Algorithm 4.1, at each step, the cyclic CD performs an exact minimization update at one coordinate (given others fixed), and the algorithm goes

through all coordinates cyclically according to a fixed ordering. In Algorithm 4.1, $\boldsymbol{E}_{ij} \in \mathbb{R}^{p\times p}$ denotes standard basis matrix with exactly one nonzero entry 1 at (i,j)-th element.

Algorithm 4.1 Cyclic CD for solving (4.13)

Input: An initialization $\hat{\boldsymbol{\Theta}}$

1: **while** not converged **do**
2: **for** each pair of $i < j$ **do**
3: $\hat{\theta}_{ij} = \hat{\theta}_{ji} \leftarrow \arg\min_{\theta_{ij}} F(\hat{\boldsymbol{\Theta}} - \hat{\theta}_{ij}\boldsymbol{E}_{ij} - \hat{\theta}_{ij}\boldsymbol{E}_{ji} + \theta_{ij}\boldsymbol{E}_{ij} + \theta_{ij}\boldsymbol{E}_{ji})$
4: **end for**
5: **for** $i = 1, 2, \ldots p$ **do**
6: $\hat{\theta}_{ii} \leftarrow \arg\min_{\theta_{ij}} F(\hat{\boldsymbol{\Theta}} - \hat{\theta}_{ii}\boldsymbol{E}_{ii} + \theta_{ii}\boldsymbol{E}_{ii})$
7: **end for**
8: **end while**

Notice that the symmetric lasso formulation in [91] is not a convex formulation as noted by [132, Lemma 2], but it is convex with respect to off-diagonals $\tilde{\boldsymbol{\Theta}}$ given the inverse of diagonals $\{\sigma^{jj}\}$. [91] propose to solve the problem by alternating minimizing $\{\sigma^{jj}\}$ and $\tilde{\boldsymbol{\Theta}}$ with the subproblem of $\tilde{\boldsymbol{\Theta}}$ solved by cyclic CD. This can be regarded as a cyclic BCD over two blocks $\mathrm{diag}(\boldsymbol{\Theta})$ and $\tilde{\boldsymbol{\Theta}}$, with another cyclic CD as the subproblem solver for $\tilde{\boldsymbol{\Theta}}$ update, which is different from Algorithm 4.1.

Cyclic CD methods are also applied in the context of best subset selection for both regression and classification settings [116, 119, 66]. There are two major differences in terms of the cyclic CD methods. First, different from the regression or classification problem, we are dealing with the sparse symmetric matrix in the covariance matrix estimation framework (4.13). The cyclic CD needs to handle the on-diagonal and (symmetric) off-diagonal entries differently, as shown in lines 3 and 6 in Algorithm 4.1. Second, the convergence guarantee of Algorithm 4.1 is unknown, even when h_{ij}'s are convex. The sparse regression problem considered in [119] is convex, smooth and component-wise strongly convex, and thus the coordinate descent enjoys a $O(1/T)$ sublinear rate of convergence [123]. However, the pseudo-likelihood framework in

(4.13) has additional log terms and quadratic-over-linear structure, which make the objective neither smooth nor componentwise strongly convex. As pointed out by [132], there is no known convergence guarantee for the cyclic CD applied to this convex formulation of symmetric lasso procedure. Later in Section 4.3.4, we will provide the convergence guarantee for Algorithm 4.1 for the root/node relaxation subproblems (4.10).

Coordinate updates: The coordinate updates in lines 3 and 6 of Algorithm 4.1 can be reduced to simple formulations. In fact, for any $i < j$ and h_{ij}, the update in line 3 of Algorithm 4.1 is equivalent to[2]

$$\hat{\theta}_{ij}^{+} = \arg\min_{\theta} \; a_{ij}\theta^2 + b_{ij}\theta + h_{ij}(\theta) \tag{4.14}$$

where

$$a_{ij} = \frac{v_j}{\hat{\theta}_{ii}} + \frac{v_i}{\hat{\theta}_{jj}}, \qquad b_{ij} = \frac{2\tilde{\boldsymbol{x}}_j^\top(\boldsymbol{r}_i - \hat{\theta}_{ij}\tilde{\boldsymbol{x}}_j)}{\hat{\theta}_{ii}} + \frac{2\tilde{\boldsymbol{x}}_i^\top(\boldsymbol{r}_j - \hat{\theta}_{ij}\tilde{\boldsymbol{x}}_i)}{\hat{\theta}_{jj}}$$

with $v_i = \tilde{\boldsymbol{x}}_i^\top \tilde{\boldsymbol{x}}_i$ and $\boldsymbol{r}_i = \tilde{\boldsymbol{X}}\hat{\boldsymbol{\theta}}_i$. The solution to (4.14) is closely related the proximal operator of h_{ij}. For all special cases we are interested in, the proximal operators can be computed in the closed form; see [116, 119] and Appendix 4.A.1 for detailed expressions, properties and their derivations.

For the diagonal entries θ_{ii}, the update in line 6 of Algorithm 4.1 is given by

$$\hat{\theta}_{ii}^{+} = \arg\min_{\theta} -\log\theta + v_i\theta + \frac{\|\boldsymbol{e}_i\|^2}{\theta} = \frac{1 + \sqrt{1 + 4v_i\|\boldsymbol{e}_i\|^2}}{2v_i}, \tag{4.15}$$

where $\boldsymbol{e}_i = \boldsymbol{r}_i - \hat{\theta}_{ii}\tilde{\boldsymbol{x}}_i$. Note that the update does not depend on the choice of h_{ij}.

In the implementation of CD, instead of computing $\boldsymbol{r}_i$'s from scratch, we store and update the values of $\boldsymbol{r}_i$'s after each coordinate update to improve the efficiency. This is known as the residual update, which is common in the coordinate descent algorithm implementation for sparse learning [89, 116].

[2]In both updates (4.14) and (4.15), we use the superscript + to disambiguate the entries after and before the coordinate update. To be more specific, the $\hat{\theta}_{ij}$ and $\hat{\theta}_{ii}$ in $a_{ij}, b_{ij}, \boldsymbol{r}_i$ and $\boldsymbol{e}_i$ are the ones before the update, while $\hat{\theta}_{ij}^{+}$ and $\hat{\theta}_{ii}^{+}$ are the ones after the update.

Active sets: The computational costs of a_{ij}, b_{ij} and $\boldsymbol{e}_i$ in the updates (4.14) and (4.15) are $O(n)$, and there are $O(p^2)$ variables in each full pass, so each iteration of Algorithm 4.1 has cost of $O(np^2)$, which becomes prohibitively expensive when n or p becomes large. To reduce the computational cost, we propose an active-set method: we run Algorithm 4.1 restricted to the diagonal variables $\mathcal{D}$ and a small subset of the off-diagonal variables $\mathcal{A} \subseteq \{(i,j) : i < j, i, j \in [p]\}$, i.e. $\boldsymbol{\Theta}|_{\mathcal{A}^c \setminus \mathcal{D}} = 0$. After solving the restricted problem, we augment the active set with the off-diagonal variables $(i,j) \in \mathcal{A}^c$ that violate the coordinate-wise optimality conditions, and resolve the problem on the new active set. We repeat this process and terminate the algorithm until there are no more violations. Such an approach is effectively used for speeding up structured-sparity learning algorithms [116, 119, 117, 52]. Our proposed method is detailed in Algorithm 4.2.

Algorithm 4.2 Active set method for solving (4.13)

Input: An intial active set $\mathcal{A}$ and initial solution $\hat{\boldsymbol{\Theta}}$

1: **while** not converged **do**

2: Get a solution for $\min_{\boldsymbol{\Theta} \in \mathbb{S}^p} \; F(\boldsymbol{\Theta}), \;\; \text{s.t.} \;\; \boldsymbol{\Theta}|_{\mathcal{A}^c \cap \mathcal{D}^c} = 0$ using Algorithm 4.1

3: $\mathcal{V} \leftarrow \{(i,j) : i < j, i, j \in [p], \hat{\theta}_{ij} = 0, 0 \notin \arg\min_{\theta_{ij}} F(\hat{\boldsymbol{\Theta}} - \hat{\theta}_{ij}\boldsymbol{E}_{ij} - \hat{\theta}_{ij}\boldsymbol{E}_{ji} + \theta_{ij}\boldsymbol{E}_{ij} + \theta_{ij}\boldsymbol{E}_{ji})\}$

4: If $\mathcal{V}$ is empty then **Terminate**; otherwise, $\mathcal{A} \leftarrow \mathcal{A} \cup \mathcal{V}$

5: **end while**

In what follows, we will discuss some details of Algorithm 4.2 when we use it to solve the root/node relaxation (4.10) and the problem for incumbents (4.7), in Sections 4.3.4 and 4.3.6, respectively.

4.3.4 Node relaxation solving

In this section, we discuss computational details as well as convergence guarantees of using Algorithms 4.1 and 4.2 introduced in Section 4.3.3 to obtain solutions to the root/node relaxation subproblems (4.10).

Coordinate updates: Recall that as a special case of the unified formulation (4.13), the node relaxation subproblem (4.10) has regularizers $h_{ij}(\theta_{ij}) = g(\theta_{ij}; \lambda_0, \lambda_2, M, \underline{z}_{ij}, \bar{z}_{ij})$. Its corresponding off-diagonal updates (4.14) in the line 3 of Algorithm 4.1 has a closed-form solution, related to the proximal operators of ψ and φ, derived in [119]. For the self-containedness of the chapter, we present these formulations and closed-form updates in Appendix 4.A.1.2.

Computational guarantee: As we mentioned earlier, due to the log-terms and quadratic-over-linear structure of the pseudolikelihood, there is no known convergence guarantee for the coordinate descent algorithm, as pointed out by [132]. The following theorem provides such convergence guarantee and presents the sublinear rate of convergence for Algorithm 4.1 applied to the relaxation subproblem (4.10).

Theorem 4.1. *Given any initialization $\mathbf{\Theta}^{(0)}$, let $\mathbf{\Theta}^{(t)}$ be the t-th iterate generated by Algorithm 4.1 (at the end of t-th while-loop), then there exists a constant C that depends on $\mathbf{\Theta}^{(0)}$, for any $t \geq 1$,*

$$F_{\mathsf{node}}(\mathbf{\Theta}^{(t)}) - F^*_{\mathsf{node}} \leq \frac{C}{t}, \tag{4.16}$$

where $F^*_{\mathsf{node}} = \min_{\mathbf{\Theta} \in \mathbb{S}^p} F_{\mathsf{node}}(\mathbf{\Theta})$.

In Appendix 4.A.2, we prove the convergence guarantee for the unified formulation (4.13) with a more class of regularizers. This also includes the equivalent convex formulation of symmetric lasso procedure with $h_{ij}(\theta) = \lambda_1|\theta|$.

Initializations: The quality of the initial active set $\mathcal{A}$ affect the number of iterations in Algorithm 4.2. Due to the similarity between the parent node and its two child nodes, we take the initial active set to be the same as the support of the relaxation solution at the parent node. For the root relaxation problem, we consider initializing the active set as the support of the warm start, obtained by the heuristic solver that will be discussed in Section 4.3.6.

Inexact solving: For the practical purpose, we do not solve the restricted problem in line 2 of Algorithm 4.2 exactly — instead we terminate Algorithm 4.1 when the

relative change in the objectives is small. In next section, we will discuss how to obtain a dual bound based on the primal inexact solution for pruning purpose.

4.3.5 Dual bounds

As mentioned before, for the practical purpose, we do not need to use Algorithm 4.2 to solve the problem exactly. Instead, inexact solutions to the node relaxations (4.10) are obtained by using Algorithm 4.2 with low accuracy. However, we still the dual bounds to perform search space pruning in BnB. We provide an efficient method to compute the dual bounds from the primal solutions. Compared to dual bounds presented in [119] for regression problem, the dual bounds derived here have several differences: (i) compared to the regression problem, our relaxation (4.10) has a quadratic-over-linear structure, an extra log terms and the symmetric constraint, so the dual becomes more complicated; (ii) we provide a unified dual expression in terms of convex conjugate functions of g, while [119] introduce two additional dual variables γ, μ for computing the dual for different regimes of $\sqrt{\lambda_0/\lambda_2}$.

We presents the Lagrangian dual of (4.10) in the following theorem:

Theorem 4.2. *A dual of Problem* (4.10) *is given by*

$$\max_{\boldsymbol{\nu}} \; D(\boldsymbol{\nu}) := p + \sum_{i=1}^{p} \log(-\|\boldsymbol{\nu}_i\|^2/4 - \tilde{\boldsymbol{x}}_i^\top \boldsymbol{\nu}_i) - \sum_{i<j} g^*(\tilde{\boldsymbol{x}}_j^\top \boldsymbol{\nu}_i + \tilde{\boldsymbol{x}}_i^\top \boldsymbol{\nu}_j; \lambda_0, \lambda_2, M, \underline{z}_{ij}, \bar{z}_{ij}), \tag{4.17}$$

where $g^*(\cdot; \lambda_0, \lambda_2, M, \underline{z}, \bar{z})$ *is the convex conjugate of* $g(\cdot; \lambda_0, \lambda_2, M, \underline{z}, \bar{z})$. *The strong duality holds, i.e.*

$$\min_{\boldsymbol{\Theta} \in \mathbb{S}^p} F_{\mathsf{node}}(\boldsymbol{\Theta}) = \max_{\boldsymbol{\nu}} D(\boldsymbol{\nu}).$$

Furthermore, if $\boldsymbol{\Theta}^*$ *is an optimal primal solution to* (4.10), *let* $\boldsymbol{r}_i^* = \tilde{\boldsymbol{X}} \boldsymbol{\theta}_i^*$ *for all* $i \in [p]$, *then* $\boldsymbol{\nu}_i^* = -2\boldsymbol{r}_i^*/\theta_{ii}^*$ *is the optimal dual solution to* (4.17).

Given any α, the convex conjugate $g^*(\alpha)$ can be computed explicitly, and we provide the computational details in Appendix 4.A.3.

Dual bounds: Let $\hat{\boldsymbol{\Theta}}$ be an inexact solution generated by Algorithm 4.1 or 4.2

applied to the node relaxation (4.10). Then, we can construct a dual solution based on $\boldsymbol{\Theta}$:

$$\hat{\boldsymbol{\nu}}_i = -2\tilde{\boldsymbol{X}}\hat{\boldsymbol{\theta}}_i/\hat{\theta}_{ii}. \tag{4.18}$$

Notice that when $-\|\hat{\boldsymbol{\nu}}_i\|^2/4 - \tilde{\boldsymbol{x}}_i^\top \boldsymbol{\nu}_i \leq 0$, the dual solution is infeasible, and thus $D(\hat{\boldsymbol{\nu}}) = -\infty$. This indicates the optimization error of the current inexact solution $\hat{\boldsymbol{\Theta}}$ is still not small enough.

However, it is possible to show the tightness of the dual bounds in a similar fashion to [119, Theorem 3], in the sense that as long as $\hat{\boldsymbol{\Theta}}$ is close to the optimal $\boldsymbol{\Theta}^*$ to (4.10) in certain metric, then $D(\boldsymbol{\Theta}^*) - D(\hat{\boldsymbol{\Theta}})$ cannot be too large, and its upper bound depends on the number of nonzero off-diagonal entries instead of $\mathcal{O}(p^2)$. The proof is omitted here because this is not the focus of the chapter.

Efficient computation of the dual bounds: A direct computation of the dual bounds $D(\hat{\boldsymbol{\nu}})$ costs $\mathcal{O}(np^2)$. This can be reduced to $\mathcal{O}(nk)$ if $\hat{\boldsymbol{\Theta}}$ is a solution from Algorithm 4.2, where k is the number of nonzero off-diagonal entries in $\hat{\boldsymbol{\Theta}}$. [119] explores the good property of ψ in the root relaxation in the sparse regression setting — translated into our setting, this is equivalent to if $\hat{\theta}_{ij} = 0$, then

$$\psi^*(\tilde{\boldsymbol{x}}_j^\top \hat{\boldsymbol{\nu}}_j + \tilde{\boldsymbol{x}}_i^\top \hat{\boldsymbol{\nu}}_i; \lambda_0, \lambda_2, M) = 0.$$

This means we compute ψ^* (as a special case of g^* in root relaxation) over the off-diagonal support of $\hat{\boldsymbol{\Theta}}$, which reduces the operations to $\mathcal{O}(nk)$.

This can be generalized to the node relaxation setting — the only difference is that if $\hat{\theta}_{ij} = 0$ and $\underline{z}_{ij} = \bar{z}_{ij} = 1$, $g^*(\tilde{\boldsymbol{x}}_j^\top \hat{\boldsymbol{\nu}}_j + \tilde{\boldsymbol{x}}_i^\top \hat{\boldsymbol{\nu}}_i; \lambda_0, \lambda_2, M, \underline{z}_{ij}, \bar{z}_{ij}) = -\lambda_0$. Since we can easily store the number of z_{ij}'s that are fixed to 1 in $\mathcal{O}(1)$ at each node, the complexity of computing dual bounds remains $\mathcal{O}(nk)$. The formal statement is presented in Proposition 4.1.

Proposition 4.1. *Let $\hat{\boldsymbol{\Theta}}$ be a solution obtained by Algorithm 4.2 applied to the node relaxation* (4.10)*, and $\hat{\boldsymbol{\nu}}$ is a dual feasible solution obtained by* (4.18)*. Denote by*

$$\hat{\mathcal{S}} = \{(i,j) : i < j, \hat{\theta}_{ij} \neq 0\}, \quad \textit{and} \quad \mathcal{F}_1 = \{(i,j) : i < j, \underline{z}_{ij} = \bar{z}_{ij} = 1\}.$$

If for any $i \in [p]$, $-\|\hat{\boldsymbol{\nu}}_i\|^2/4 - \tilde{\boldsymbol{x}}_i^\top \hat{\boldsymbol{\nu}}_i > 0$, *then*

$$D(\hat{\boldsymbol{\nu}}) = p + \sum_{i=1}^{p} \log(-\|\hat{\boldsymbol{\nu}}_i\|^2/4 - \tilde{\boldsymbol{x}}_i^\top \hat{\boldsymbol{\nu}}_i) + \lambda_0 |\mathcal{F}_1 \backslash \hat{\mathcal{S}}| \\ - \sum_{(i,j)\in\hat{\mathcal{S}}} g^*(\tilde{\boldsymbol{x}}_j^\top \hat{\boldsymbol{\nu}}_i + \tilde{\boldsymbol{x}}_i^\top \hat{\boldsymbol{\nu}}_j; \lambda_0, \lambda_2, M, \underline{z}_{ij}, \bar{z}_{ij}).$$

Otherwise, $D(\hat{\boldsymbol{\nu}}) = -\infty$.

In practice, we always make sure $\mathcal{F}_1$ is a subset of the active set $\mathcal{A}$, and thus both $\hat{\mathcal{S}}$ and $\mathcal{F}_1$ are subsets of $\mathcal{A}$. We can compute the convex conjugate terms restricted to $\mathcal{A}$, and the corresponding computational cost is $\mathcal{O}(n|\mathcal{A}|)$.

4.3.6 Heuristic solver and incumbents

In this section, we discuss computational details of our heuristic solver to solve (4.7), using Algorithms 4.1 and 4.2 introduced in Section 4.3.3. We will first present the coordinate updates corresponding to Problem (4.7), and then we discuss choices of $\mathcal{S}$ and the initializations for both initial incubment solving and incumbent solving at each node.

Coordinate updates: We notice that the objective in (4.7) is a special case of the unified formulation (4.13) with

$$h_{ij}(\theta_{ij}) = \begin{cases} \lambda_0 \mathbf{1}\{\theta_{ij} \neq 0\} + \lambda_2 \theta_{ij}^2 + \chi\{|\theta_{ij}| \leq M\}, & \text{if } (i,j) \in \mathcal{S} \\ \chi\{\theta_{ij} = 0\}, & \text{if } (i,j) \in \mathcal{S}^c \end{cases}.$$

Its corresponding off-diagonal update in (4.14) in the line 3 of Algorithm 4.1 has a closed-form expression. When $(i,j) \in \mathcal{S}$ and $M = \infty$, the corresponding expression is derived in [116], and we extend it to the big-M constraint setting and present the results in Appendix 4.A.1.3.

Heuristics of choosing $\mathcal{S}$: For the initial incumbent solving, since we do not have prior knowledge of the support of the problem, we take $\mathcal{S}$ to be the set of all pairs, i.e. $\mathcal{S} = \{(i,j) : 1 \leq i < j \leq p\}$, and we solve the problem by Algorithm 4.2.

For the incumbent solving at each node, we attempt to improve the upper bound based on the solution $\hat{\boldsymbol{\Theta}}$ at the current node's relaxation, and thus we set $\mathcal{S}$ based on $\boldsymbol{z}$ induced by $\hat{\boldsymbol{\Theta}}$. We propose two options—(i) directly taking the support of $\boldsymbol{z}$, i.e. $\mathcal{S} = \{(i,j) : i < j, z_{ij} > 0\}$; (ii) taking the support of rounded $\boldsymbol{z}$, i.e. $\mathcal{S} = \{(i,j) : i < j, z_{ij} \geq 0.5\}$. In this case, due to the sparsity of $\hat{\boldsymbol{\Theta}}$, we expect $\mathcal{S}$ to be small, so Algorithm 4.1 should be efficient enough to solve the problem.

Initializations: Since (4.7) is a discrete nonconvex problem, so the number of iterations in Algorithm 4.1 or 4.2 and the quality of the approximate solution given by the algorithms are affected by the quality of the initial solution $\hat{\boldsymbol{\Theta}}$ and/or the quality of the initial active set $\mathcal{A}$.

For initial incumbent solving, as we do not have any prior knowledge, we initialize Algorithm 4.2 with the trivial solution

$$\hat{\boldsymbol{\Theta}}^{(0)} = \mathrm{diag}(v_1^{-1}, \ldots, v_p^{-1}),$$

which is optimal when all the off-diagonal entries are forced to be 0. We obtain the initial active set $\mathcal{A}$ by correlation screening [116] — computing the correlation matrix of $\boldsymbol{X}$ and taking a small portion of coordinates (i,j) that have highest correlations in each row.

For the incumbent solving at every node, we initialize Algorithm 4.1 with the current relaxation solution $\hat{\boldsymbol{\Theta}}$ restricted on $\mathcal{S}$.

4.4 Statistical Properties

In this section, we investigate the statistical properties of the estimator (4.4). We consider two different criteria for our estimator. for our analysis. First, we present estimation error bounds of the form $\|\boldsymbol{\Theta}^* - \hat{\boldsymbol{\Theta}}\|_F$ where $\boldsymbol{\Theta}^*$ is the underlying precision matrix and $\hat{\boldsymbol{\Theta}}$ is the estimated one. In the high-dimensional regime where p/n can be large, we consider the variable selection properties of our estimator.

4.4.1 Estimation Error Bound

Before proceeding with our results in this section, we state our assumptions on the model for this case.

Assumption 4.1. *Suppose the GGM model* (4.1) *holds. Let* β^*_{ij}, σ^*_j *be as defined in* (4.3)*. We assume:*

(A1) *There exist* $l_\sigma, u_\sigma \geq 0$ *such that for any* $j \in [p]$, $l_\sigma \leq \sigma^*_j \leq u_\sigma$ *with* $u_\sigma \geq 1$.

(A2) *For any* $i \neq j$, $|\beta^*_{ij}| \lesssim 1$.

(A3) *We have* $l^2_\sigma \geq \frac{6}{25}u^4_\sigma + \frac{2}{5}u^2_\sigma$.

(A4) *For* $j \in [p]$, $\left|\left\{i \in [p] : i \neq j, \beta^*_{ij} \neq 0\right\}\right| \leq k$.

(A5) *For the matrix* $\mathbf{\Theta}^*$*, we assume*

$$\min_{\substack{S \subseteq [p] \\ |S| \leq 2k}} \lambda_{\min}(\mathbf{\Sigma}^*_{S,S}) \geq \kappa^2 \gtrsim 1$$

where $\mathbf{\Sigma}^* = (\mathbf{\Theta}^*)^{-1}$ *and* κ *is an absolute constant.*

Assumptions **(A1)** to **(A3)** stated above ensure that the matrix $\mathbf{\Theta}^*$ is normalized and does not have large or small entries. Assumption **(A4)** states that each column of $\mathbf{\Theta}^*$ is sparse and off-diagonals of each column have at most k nonzeros. This is a standard assumption in the GGM literature [213, Chapter 11]. Assumption **(A5)** states that the sub-matrices of $\mathbf{\Sigma}^*$ are not badly conditioned. This assumption is required in our analysis to be able to derive estimation error bounds. In our analysis, we consider κ to be a fixed absolute constant while other parameters can vary.

In light of Assumption **(A1)**, we consider a slightly modified version of Problem (4.4). Namely, we consider

$$\begin{aligned} \min_{\beta,\sigma} \quad & \sum_{j=1}^{p} \left[\log(\sigma_j) + \frac{1}{n}\frac{1}{2\sigma^2_j} \left\| \boldsymbol{x}_j - \sum_{i:i\neq j} \beta_{ij}\boldsymbol{x}_i \right\|^2_2 \right] \\ \text{s.t.} \quad & l_\sigma \leq \sigma_j \leq u_\sigma,\ j \in [p]; \quad \beta_{ij}\sigma^2_i = \beta_{ji}\sigma^2_j,\ i \neq j, \quad \beta_{ii} = 0,\ i \in [p] \\ & |\{i \in [p] : i \neq j, \beta_{ij} \neq 0\}| \leq k, j \in [p] \end{aligned} \tag{4.19}$$

where we add additional boundedness constraints on σ_j. Theorem 4.3 establishes an estimation error bound for Problem (4.19).

Theorem 4.3. *Let $\{\hat{\beta}_{ij}\}, \{\hat{\sigma}_j\}$ be the optimal solution to Problem* (4.19)*. Under Assumptions **(A1)** to **(A5)** with $p/k > 5$, if $n \gtrsim k \log p$,*

$$\sum_{j\in[p]}(\hat{\sigma}_j - \sigma_j^*)^2 + \frac{1}{u_\sigma^2}\sum_{j\in[p]}\sum_{i:i\neq j}(\beta_{ij}^* - \hat{\beta}_{ij})^2 \lesssim \frac{u_\sigma^2 kp\log(2p/k)}{l_\sigma^2 n} \tag{4.20}$$

with high probability.[3]

Theorem 4.3 established an ℓ_2 error bound on the estimation of coefficients β^* and variances σ^*. As our goal is to estimate the precision matrix $\mathbf{\Theta}^*$, we perform a transformation on the results of Problem (4.19), based on (4.3). Theorem 4.4 below establishes the estimation error bound on the precision matrix.

Theorem 4.4. *Let $\{\hat{\beta}_{ij}\}, \{\hat{\sigma}_j\}$ be the optimal solution to Problem* (4.19)*, and let $\hat{\theta}_{jj} = \frac{1}{\hat{\sigma}_j^2}$ and $\hat{\theta}_{ji} = -\frac{\hat{\beta}_{ij}}{\hat{\sigma}_j^2}$ for $i \neq j \in [p]$. Then, under the assumptions of Theorem 4.3,*

$$\left\|\hat{\mathbf{\Theta}} - \mathbf{\Theta}^*\right\|_F^2 \lesssim \frac{(u_\sigma^6 + u_\sigma^8)kp\log(2p/k)}{l_\sigma^{10} n} \tag{4.21}$$

with high probability.[3]

Remark 4.1 (Comparison with statistically optimal results)**.** *Theorem 4.4 shows that our proposed estimator achieves a Frobenius estimation rate of $\sqrt{kp\log p/n}$. This rate typically matches the estimation rate of current methods for GGM and is known to be minimax optimal up to logarithmic factors (see [194] for a detailed discussion on estimation rate for GGM).*

Remark 4.2 (Comparison with prior pseudo-likelihood-based methods)**.** *To our knowledge, Theorem 4.4 is the first result that presents a non-asymptotic estimation guarantee for pseudo-likelihood-based GGM that results in a symmetric estimator. GGM estimators based on pseudo-likelihood have been considered by [183, 132, 91] however, none of these papers provide a non-asymptotic analysis similar to the one in*

[3] An explicit expression for probability can be found in (4.78).

Theorem 4.4. (The analysis of [183] is asymptotic as $n \to \infty$). The method of [166] is based on solving a sparse linear regression-type problem. However, their analysis is focused on variable selection (which we discuss later). In addition, their estimator does not enforce symmetry, unlike us.

An important property of our estimator is symmetry. Although the bound of Theorem 4.4 is minimax optimal, it does not justify how a symmetric solution improves the estimation accuracy. As a result, below we consider an illustrative example in which we quantify the benefit of symmetry constraints.

Example 4.1. *Let $p = 2$, $c \in (0, 1)$, and*

$$\boldsymbol{\Theta}^* = \begin{bmatrix} 1 & c \\ c & 1 \end{bmatrix}, \quad \boldsymbol{\Sigma}^* = (\boldsymbol{\Theta}^*)^{-1} = \frac{1}{1-c^2} \begin{bmatrix} 1 & -c \\ -c & 1 \end{bmatrix} \tag{4.22}$$

and data is generated per model (4.1). As we are interested to investigate the effect of symmetry on values of β, we assume the variance values $(\sigma_j^*)^2$ are known. As a result, in the symmetric case, we consider the problem:

$$\min_{\beta_1, \beta_2} \quad \|\boldsymbol{x}_1 - \beta_1 \boldsymbol{x}_2\|_2^2 + \|\boldsymbol{x}_2 - \beta_2 \boldsymbol{x}_1\|_2^2 \quad s.t. \quad \beta_1 = \beta_2. \tag{4.23}$$

The optimal solution to (4.23) is given as

$$\hat{\beta}_1 = \hat{\beta}_2 = \hat{\beta} = \frac{2\boldsymbol{x}_1^\top \boldsymbol{x}_2}{\boldsymbol{x}_1^\top \boldsymbol{x}_1 + \boldsymbol{x}_2^\top \boldsymbol{x}_2}$$

resulting in the total estimation error

$$2\left(\beta^* - \hat{\beta}\right)^2 = 2\left(c + \frac{2\boldsymbol{x}_1^\top \boldsymbol{x}_2}{\boldsymbol{x}_1^\top \boldsymbol{x}_1 + \boldsymbol{x}_2^\top \boldsymbol{x}_2}\right)^2. \tag{4.24}$$

Without the symmetry constraints, we consider

$$\min_{\beta_1, \beta_2} \quad \|\boldsymbol{x}_1 - \beta_1 \boldsymbol{x}_2\|_2^2 + \|\boldsymbol{x}_2 - \beta_2 \boldsymbol{x}_1\|_2^2 \tag{4.25}$$

which leads to solutions

$$\hat{\beta}_1 = \frac{\boldsymbol{x}_1^\top \boldsymbol{x}_2}{\boldsymbol{x}_2^\top \boldsymbol{x}_2}, \hat{\beta}_2 = \frac{\boldsymbol{x}_1^\top \boldsymbol{x}_2}{\boldsymbol{x}_1^\top \boldsymbol{x}_1}.$$

As a result, the total estimation error in this case is

$$\left(\beta^* - \hat{\beta}_1\right)^2 + \left(\beta^* - \hat{\beta}_2\right)^2 = \left(c + \frac{\boldsymbol{x}_1^\top \boldsymbol{x}_2}{\boldsymbol{x}_1^\top \boldsymbol{x}_1}\right)^2 + \left(c + \frac{\boldsymbol{x}_1^\top \boldsymbol{x}_2}{\boldsymbol{x}_2^\top \boldsymbol{x}_2}\right)^2. \tag{4.26}$$

As it can be seen, the estimation error in the asymmetric case depends on quantities $\boldsymbol{x}_1^\top \boldsymbol{x}_1$ *and* $\boldsymbol{x}_2^\top \boldsymbol{x}_2$ *separately while the error in the symmetric case only depends on* $\boldsymbol{x}_1^\top \boldsymbol{x}_1 + \boldsymbol{x}_2^\top \boldsymbol{x}_2$*. Intuitively, the latter has an averaging effect on the error caused by randomness of the data. In other words,* $\boldsymbol{x}_1^\top \boldsymbol{x}_1 + \boldsymbol{x}_2^\top \boldsymbol{x}_2$ *has a lower variance compared to* $\boldsymbol{x}_1^\top \boldsymbol{x}_1$ *and* $\boldsymbol{x}_2^\top \boldsymbol{x}_2$ *separately. This leads to lower estimation error and better estimation performance in the symmetric case.*
Mathematically, one can show (see Appendix 4.C for details) for the symmetric case, the estimation error is upper bounded as

$$\mathbb{P}\left(2\left(\beta^* - \hat{\beta}\right)^2 \leq \frac{4\epsilon^2(c^2+1)}{(1-\epsilon)^2}\right) \geq 1 - \frac{2+2c^2}{n\epsilon^2}$$

for $\epsilon \in (0,1)$*. On the other hand, for the asymmetric case we have*

$$\mathbb{P}\left(\left(\beta^* - \hat{\beta}_1\right)^2 + \left(\beta^* - \hat{\beta}_2\right)^2 \leq \frac{4\epsilon^2(c^2+1)}{(1-\epsilon)^2}\right) \geq 1 - \frac{5+c^2}{n\epsilon^2}.$$

This shows for a given ϵ*, the symmetric case can provide the same error bound with a higher probability or for a fixed confidence level, the symmetric case can provide a lower error.*

4.4.2 Support Recovery Guarantees

In this section, we analyze our estimator from variable selection point of view and present support recovery guarantees. Before proceeding with our results, we introduce an adjusted version of our estimator that is more suited to high-dimensional settings where variable selection is canon. As we seek to estimate the support, we relax

the symmetry constraint (4.4b) so that the support of β_{ij} is symmetric, that is, the location of nonzeros of $\{\beta_{ij}\}$ is symmetric. Moreover, as we show in this case it is not required to know the value of sparsity k so we relax constraint (4.4c) as a penalty. As a result, the estimator we consider is given as

$$\begin{aligned} \min_{\beta,\sigma,z} \quad & \sum_{j=1}^{p} \left[\log(\sigma_j) + \frac{1}{2n\sigma_j^2} \left\| \boldsymbol{x}_j - \sum_{i:i\neq j} \beta_{ij} \boldsymbol{x}_i \right\|_2^2 \right] + \lambda \sum_{i\neq j} z_{ij} \\ \text{s.t.} \quad & z_{ij} \in \{0,1\}, \quad z_{ij} = z_{ji}, \quad z_{ij} = 0 \Rightarrow \beta_{ij} = 0, \quad \beta_{ii} = 0 \quad i \neq j \in [p] \\ & \sigma_j \geq \sqrt{\ell}, j \in [p] \end{aligned} \tag{4.27}$$

where z_{ij} controls the sparsity structure of $\{\beta_{ij}\}$, similar to Problem (4.6). In this section, we use the following assumptions.

Assumption 4.2. *Suppose the GGM model* (4.1) *holds. Let* β_{ij}^* *be as defined in* (4.3). *We assume:*

(B1) *There exist* $u_\sigma \geq l_\sigma > 0$ *such that for any* $j \in [p]$, $l_\sigma \leq \sigma_j^* \leq u_\sigma$ *and* $u_\sigma \leq 5l_\sigma$.

(B2) *For* $i, j \in [p]$, $i \neq j$, *we have* $|\beta_{ij}^*| \leq 1/\sqrt{k}$.

(B3) *For* $i.j \in [p]$, *we have*

$$(\boldsymbol{\Sigma}^*)_{ij} \lesssim 1$$

and

$$\max_{j\in[p]} \frac{(\boldsymbol{\Sigma}^*)_{jj}}{(\sigma_j^*)^2} \leq \frac{400}{7}$$

where $\boldsymbol{\Sigma}^* = (\boldsymbol{\Theta}^*)^{-1}$.

(B4) *There exists a value* $\beta_{\min}$ *such that* $\beta_{\min} \geq \sqrt{\frac{\eta \log p}{n}}$ *for some sufficiently large numerical constant* $\eta \gtrsim u_\sigma^2$, *and*

$$|\beta_{ij}^*| \geq \beta_{\min} \text{ for all } (i,j) \in [p] \times [p] \text{ such that } \beta_{ij}^* \neq 0 \text{ and } i > j. \tag{4.28}$$

(B5) *For* $j \in [p]$, $\left|\left\{i \in [p] : i \neq j, \beta_{ij}^* \neq 0\right\}\right| \leq k$ *for some* $k > 0$.

(B6) *For the matrix $\mathbf{\Theta}^*$, we assume*

$$3 \geq \max_{\substack{S \subseteq [p] \\ |S| \leq 2k}} \lambda_{\max}(\mathbf{\Sigma}^*_{S,S}) \geq \min_{\substack{S \subseteq [p] \\ |S| \leq 2k}} \lambda_{\min}(\mathbf{\Sigma}^*_{S,S}) \geq \kappa^2 > 0.3$$

where κ is an absolute constant.

In Assumptions **(B1)** to **(B3)**, we generally assume that $\mathbf{\Theta}^*, \mathbf{\Sigma}^*$ are bounded. Assumption **(B4)** is a non-degeneracy condition that is generally needed to achieve support recovery. Such assumptions are common in the literature [218]. Assumption **(B5)** is the sparsity assumption on the underlying model. Note that the value of k does not appear in Problem (4.27). Finally, Assumption **(B6)** is a condition number assumption that is generally common in the literature. Theorem 4.5 presents support recovery guarantees for our estimator.

Theorem 4.5. *Suppose Assumptions **(B1)** to **(B6)** hold. Let $\{\hat{z}_{ij}\}$ be the optimal solution to Problem* (4.27) *with $\ell = l_\sigma/\sqrt{3}$ and $\{z^*_{ij}\}$ be the binary matrix corresponding to the correct support, such that $z^*_{ij} = 1 \Leftrightarrow \theta^*_{ij} \neq 0$ for $i \neq j$. Then, $\hat{z}_{ij} = z^*_{ij}$ for $i \neq j \in [p]$ with high probability[4] if $n = c_n k \log p$ and $\lambda = c_\lambda \log p/n$ for some sufficiently large absolute constants $c_n, c_\lambda > 0$.*

Remark 4.3 (Comparison with statistically optimal results)**.** *We note that the number of samples $n \gtrsim k \log p$ required in Theorem 4.5 for correct support recovery is Minimax optimal up to logarithmic constants [218] and matches the support recovery results of current methods (for example, see [45, 90].)*

Remark 4.4 (The effect of symmetry)**.** *As seen in Theorem 4.5, to achieve perfect support recovery, we require a non-degeneracy condition as given by the $\beta_{\min}$ assumption **(B4)**. However, we note that as Problem* (4.27) *results in a symmetric support, we need non-degeneracy conditions only on half of the value of β^*_{ij} as stated in Assumption **(B4)**. Intuitively, an error in estimating the support propagates to at least one other location (due to the symmetric support), meaning only half of the β^*_{ij} coefficients need to be non-degenerate. To our knowledge, other estimators based on linear*

[4] An explicit expression for the probability can be found in (4.121)

regression (such as [163, 166]) do not enjoy this property, as they do not enforce symmetry in their linear regression-based estimators.

4.5 Numerical Experiments

In this section, we present various numerical experiments to compare our proposed method against several competing methods in terms of computational efficiency, statistical performance and a downstream task of portfolio optimization.

Competing Methods: We compare our method to following algorithms for graphical models: `GLASSO` [90], `CONCORD` [132] and `CLIME` [45]. We use a validation set to select parameters for different methods.

4.5.1 Synthetic Data

In this section, we investigate the computational and statistical performance of our proposed estimator using synthetic datasets. The data points $\boldsymbol{x}^{(i)}$ for $i = 1, \cdots, n$ are drawn independently from the normal distribution $\mathcal{N}(\mathbf{0}, (\boldsymbol{\Theta}^*)^{-1})$. We also draw n validation points from the same distribution. We consider two scenarios for the precision matrix $\boldsymbol{\Theta}^* \in \mathbb{R}^{p \times p}$ as outlined below:

1. **Uniform Sparsity:** We let $\boldsymbol{\Theta} = \boldsymbol{B} + \delta \boldsymbol{I}_p$ where $\boldsymbol{B}$ is generated as follows. Each entry of $\boldsymbol{B}$ is set to 0.5 with probability p_0 and zero with probability $1 - p_0$. We also symmetrize $\boldsymbol{B}$ as $(\boldsymbol{B} + \boldsymbol{B}^\top)/2$. Then, the value of δ is chosen so that the condition number of $\boldsymbol{B} + \delta \boldsymbol{I}_p$ is as desired. Finally, $\boldsymbol{\Theta}^{-1}$ is normalized so that each variable has unit variance. We set $p_0 = k/2p$. Note that $\boldsymbol{\Theta}$ has kp nonzero entries on average.

2. **Banded Precision:** We let $\boldsymbol{\Theta} = \boldsymbol{B} + \delta \boldsymbol{I}_p$ where $\boldsymbol{B}$ is as

$$b_{ij} = \begin{cases} 0 & \text{if } |i-j| > k/2 \text{ or } i = j \\ 0.5^{|i-j|} & \text{if } |i-j| \leq k/2. \end{cases}$$

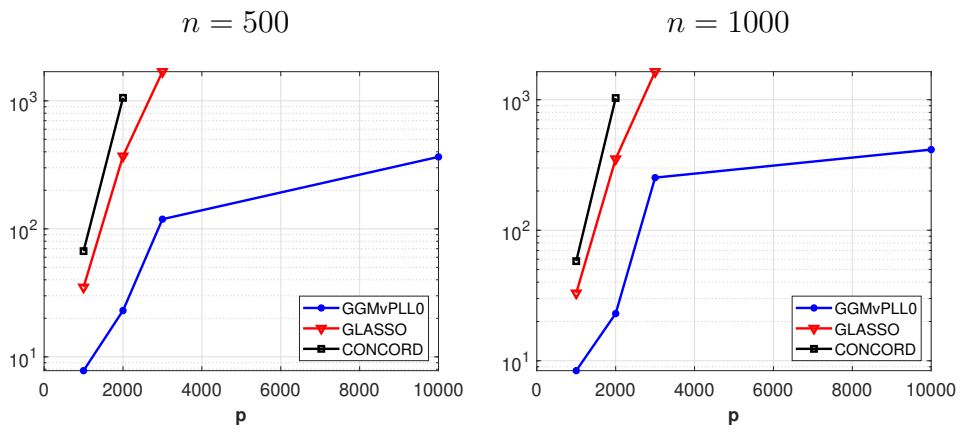

Figure 4-1: Runtimes (in seconds) for different estimators in Section 4.5.1.1.

where k is the bandwidth. Then, the value of δ is chosen so that the condition number of $\boldsymbol{B}+\delta\boldsymbol{I}_p$ is as desired. Finally, $\boldsymbol{\Theta}^{-1}$ is normalized so that each variable has unit variance. Note that $\boldsymbol{\Theta}$ has $k+1$ nonzeros per column.

The results reported here are the averages of 10 independent runs.

4.5.1.1 Timing benchmarks

In this section, we compare the runtime of our method to other (convex) estimators. We use the uniform sparsity scenario from above. We set the condition number of $\boldsymbol{\Theta}^*$ to $p/20$ and $k = 10$. Based on our experiment, `GLASSO` and `CONCORD` provide the best runtime overall and we compare the runtime of our estimator to them. The results for different values of parameters are shown in Figure 4-1. The experiments are done on a personal computer equipped with AMD Ryzen 9 5900X CPU and 32GB of RAM. In these examples, our BnB framework achieves average optimality gaps less than 2%. We also use warm-starts for `GLASSO` as this leads to faster convergence of this method.

As it can be seen, our framework is the fastest estimator compared to `GLASSO` and `CONCORD`. In addition, in our experiments `CLIME` did not scale to experiments considered in Figure 4-1. Our method scales to $p = 10^4$, while in our experiments, `GLASSO` only scales to $p \approx 3000$ and `CONCORD` only scales to $p \approx 2000$ variables. This is while `GGMvPLL0` is almost an order of magnitude faster than the aforementioned

estimators for $p \leq 3000$.

4.5.1.2 Statistical benchmarks

In this section, we use synthetic datasets to compare the statistical performance of our estimator to other algorithms. We set $p = 200$ and consider the following scenarios. In terms of performance metrics, we report the normalized estimation error $\|\hat{\mathbf{\Theta}} - \mathbf{\Theta}^*\|_F / \|\mathbf{\Theta}^*\|_F$ where $\mathbf{\Theta}^*$ is the true precision matrix and $\hat{\mathbf{\Theta}}$ is the estimated one. Next, we report Matthews Correlation Coefficient (MCC) which is defined as

$$\text{MCC} = \frac{\text{TP} \times \text{TN} - \text{FP} \times \text{FN}}{\sqrt{(\text{TP} + \text{FP})(\text{TP} + \text{FN})(\text{TN} + \text{FP})(\text{TN} + \text{FN})}}$$

where

$$\text{TP} = |\{(i,j) : \theta^*_{ij}, \hat{\theta}_{ij} \neq 0\}|, \text{FP} = |\{(i,j) : \theta^*_{ij} = 0, \hat{\theta}_{ij} \neq 0\}|$$
$$\text{TN} = |\{(i,j) : \theta^*_{ij}, \hat{\theta}_{ij} = 0\}|, \text{FN} = |\{(i,j) : \theta^*_{ij} \neq 0, \hat{\theta}_{ij} = 0\}|.$$

Note that a higher value of MCC implies a better support recovery performance. Finally, we report the support size of each estimator as $\text{NNZ} = |\{(i,j) : \hat{\theta}_{ij} \neq 0\}|$.

Scenario 1, Banded Precision: In this setup, we let $n = 50, \cdots, 300$ and set the condition number to 100, and the sparsity to $k = 6$. Here we compare the outcomes of different methods. The results for this case are shown in Figure 4-2. As it can be seen, our proposed estimator provides the lowest estimation error, and the highest MCC (which implies the best support recovery), while leading to a sparse solution. Although `CONCORD` provides good support recovery, it leads to bad estimation performance. `GLASSO` provides good estimation performance, however, similar to `CLIME`, leads to many false positives and larger support sizes, resulting in poor support recovery performance.

Scenario 2, Uniform Sparsity: In this setup, we let $n = 50, \cdots, 300$ and set the condition number to 200, and the sparsity to $k = 5, 10$. The results for $k = 5$ are shown in Figure 4-3 and the results for $k = 10$ can be found in Figure 4-4. Overall, as it can be seen our proposed estimator provides good estimation and support recovery performance. Moreover, our estimator is sparse, specially compared

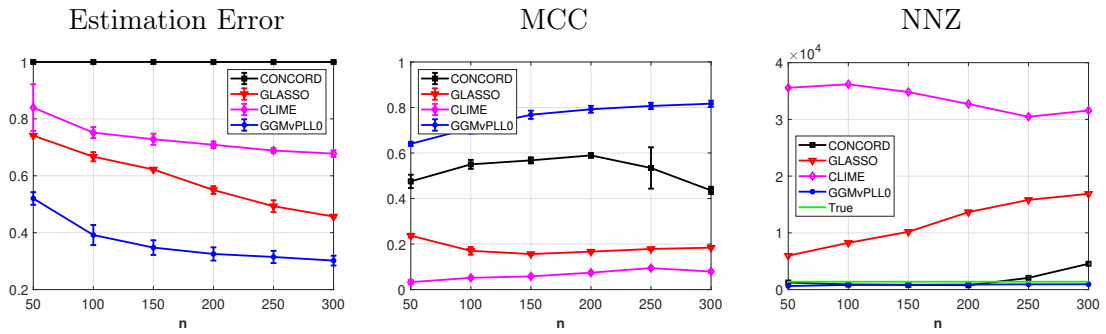

Figure 4-2: Comparison for the banded precision model in Section 4.5.1.2 with $k = 6$.

to `CLIME` and `GLASSO`. Another observation is that increasing the sparsity level leads to worse statistical performance, which is expected.

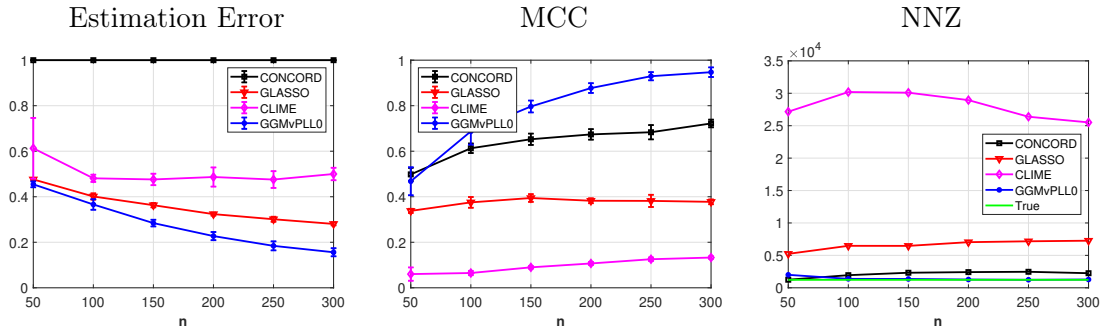

Figure 4-3: Comparison for the uniforms sparsity model in Section 4.5.1.2 with $k = 5$ and $p = 200$.

Finally, we consider a case to investigate the statistical properties of GGMs in high-dimensional settings. To this end, we set $p = 3000$, $k = 10$ and we let the condition number to be 150. As discussed in Section 4.5.1.1, only our method and `GLASSO` scale to these data instances. The results for this case are shown in Figure 4-5. As it can be seen, `GGMvPLL0` leads to almost-perfect support recovery for $n \approx 1000$ while providing better estimation performance compared to `GLASSO`. Moreover, `GLASSO` leads to numerous false positives and a dense support, as observed before.

4.5.2 Financial application

In this section, we consider portfolio optimization as a financial application of GGM. We use the stocks returns data extracted from Yahoo Finance from 2005 to 2019

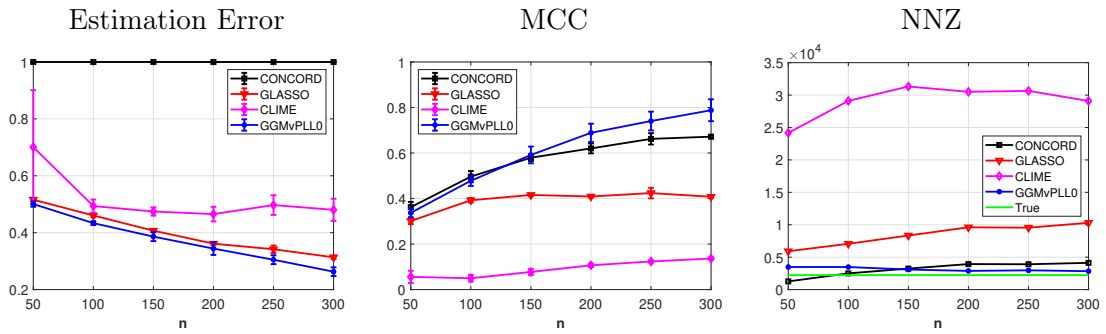

Figure 4-4: Comparison for the uniforms sparsity model in Section 4.5.1.2 with $k = 10$ and $p = 200$.

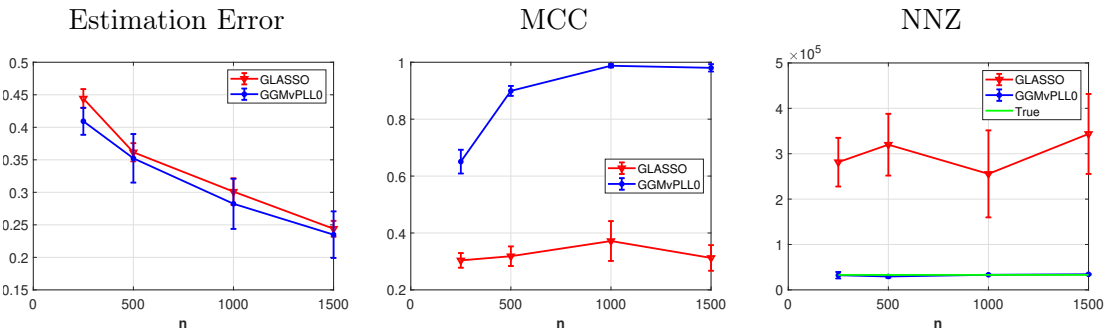

Figure 4-5: Comparison for the uniforms sparsity model in Section 4.5.1.2 with $k = 10$ and $p = 3000$.

for 1452 companies. Given the data, the goal of portfolio optimization is to select a portfolio such that leads to maximum returns and minimum risk over the portfolio [157]. Given the returns data matrix $\boldsymbol{X} \in \mathbb{R}^{n \times p}$ and a portfolio $\boldsymbol{w} \in \mathbb{R}^p_{\geq 0}$ such that $\sum_{i=1}^p w_i = 1$, the values of returns and risk are defined as

$$\begin{aligned} r &= \sum_{i=1}^{n} (\boldsymbol{X}\boldsymbol{w})_i \\ \sigma &= \sqrt{\mathrm{VAR}(\boldsymbol{X}\boldsymbol{w})}, \end{aligned} \tag{4.29}$$

respectively, where VAR denotes the variance of the vector. To select the optimal portfolio, we solve the quadratic portfolio selection problem:

$$\min_{\boldsymbol{w}} \quad \boldsymbol{w}^\top \boldsymbol{\Sigma}_X \boldsymbol{w} \quad \text{s.t.} \quad \boldsymbol{w} \in \mathbb{R}^p_{\geq 0}, \quad \sum_{i=1}^{p} w_i = 1 \tag{4.30}$$

	GGMvPLLO	GLASSO	CONCORD	CLIME
Returns	25.02	24.98	24.87	24.50
Risk	0.38	0.34	0.41	0.47
$\|\hat{\boldsymbol{\Theta}}\|_0$	2398	3060	107	3369
Runtime	19.02	0.42	0.39	44.11

Table 4.1: Simulation results for the real dataset in Section 4.5.2

where $\boldsymbol{\Sigma}_X$ is an estimation of the covariance matrix of the data. To obtain a consistent estimation of $\boldsymbol{\Sigma}_X$, we run different GGM methods on $\boldsymbol{X}$ to achieve $\hat{\boldsymbol{\Theta}}$ as an estimation of $\boldsymbol{\Sigma}_X^{-1}$, and use $\hat{\boldsymbol{\Theta}}^{-1}$ as the covariance matrix. Then, after selecting the portfolio by solving (4.30), we calculate the returns and risk on a held-out test set of data points. For more details on the setup, see [132].

We consider two cases. In the first case, we select the top 100 stocks with highest variance over time. Then, use GGM methods to estimate $\boldsymbol{\Sigma}_X$ and select the optimal portfolio using 1000 training data points and 500 validation points. Then, we use 1000 test data points to calculate the returns and risk. The average results for 20 selections of train/validation/test data are reported in Table 4.1. Overall, we see that our method provides the highest return. In terms of risk, `GLASSO` has a lower risk compared to our method, however, our method leads to lower risk compared to other methods. This is while our estimator is more sparse than `GLASSO`. In comparison to `CONCORD`, our method is more dense but has higher returns and lower risk. Overall, our method is performing well both statistically and computationally.

Next, we use every stock in the dataset and repeat the same experiment. In this case, only our method and `GLASSO` provide meaningful results, reported in Table 4.2. Overall, our method provides a considerably higher value of return, while providing better returns to risk ratio. This is while our solution is more sparse and our algorithm is faster, showing that our estimator is superior in terms of statistical and computational performance.

	GGMvPLLO	GLASSO
Returns	8.96	2.50
Risk	0.36	0.20
$\|\hat{\mathbf{\Theta}}\|_0$	27055	114450
Runtime	459	1470

Table 4.2: Simulation results for the real dataset in Section 4.5.2

4.A Results related to computations

4.A.1 Properties and optimization oracles related to regularizers

In this subsection, we present properties and optimization oracles related to regularizers, for the computations in the coordinate descent updates (4.14) and (4.15).

In Sections 4.A.1.1 and 4.A.1.4, we first present the derivations related to updates (4.14) and (4.15), and we reduce the off-diagonal update (4.15) to a proximal operator computation problem. In Sections 4.A.1.2 and 4.A.1.3, we derive the closed-form expressions of the proximal operators for the convex regularizer g (4.11) in the node/root relaxation subproblem (4.10), and the L_0L_2 regularizers in the incumbent solving problem (4.7).

4.A.1.1 Off-diagonal update

We show that the update of $\hat{\theta}_{ij}$ in line 3 of Algorithm 4.1 is given by (4.14). For any $i < j$ and h_{ij}, we have

$$
\begin{aligned}
& F(\hat{\boldsymbol{\Theta}} - \hat{\theta}_{ij}\boldsymbol{E}_{ij} - \hat{\theta}_{ij}\boldsymbol{E}_{ji} + \theta_{ij}\boldsymbol{E}_{ij} + \theta_{ij}\boldsymbol{E}_{ji}) \\
= & \text{ Const} + \frac{1}{\hat{\theta}_{ii}}\|\tilde{\boldsymbol{X}}\hat{\boldsymbol{\theta}}_i - \hat{\theta}_{ij}\tilde{\boldsymbol{x}}_j + \theta_{ij}\tilde{\boldsymbol{x}}_j\|^2 + \frac{1}{\hat{\theta}_{jj}}\|\tilde{\boldsymbol{X}}\hat{\boldsymbol{\theta}}_j - \hat{\theta}_{ij}\tilde{\boldsymbol{x}}_i + \theta_{ij}\tilde{\boldsymbol{x}}_i\|^2 + h_{ij}(\theta_{ij}) \\
\overset{(a)}{=} & \text{ Const} + \frac{1}{\hat{\theta}_{ii}}\|\boldsymbol{r}_i - \hat{\theta}_{ij}\tilde{\boldsymbol{x}}_j + \theta_{ij}\tilde{\boldsymbol{x}}_j\|^2 + \frac{1}{\hat{\theta}_{jj}}\|\boldsymbol{r}_j - \hat{\theta}_{ij}\tilde{\boldsymbol{x}}_i + \theta_{ij}\tilde{\boldsymbol{x}}_i\|^2 + h_{ij}(\theta_{ij}) \\
\overset{(b)}{=} & \text{ Const} + \frac{\|\tilde{\boldsymbol{x}}_j\|^2}{\hat{\theta}_{ii}}\theta_{ij}^2 + \frac{2\tilde{\boldsymbol{x}}_j^\top(\boldsymbol{r}_i - \hat{\theta}_{ij}\tilde{\boldsymbol{x}}_j)}{\hat{\theta}_{ii}}\theta_{ij} + \frac{\|\tilde{\boldsymbol{x}}_i\|^2}{\hat{\theta}_{jj}}\theta_{ij}^2 + \frac{2\tilde{\boldsymbol{x}}_i^\top(\boldsymbol{r}_j - \hat{\theta}_{ij}\tilde{\boldsymbol{x}}_i)}{\hat{\theta}_{jj}}\theta_{ij} + h_{ij}(\theta_{ij}) \\
\overset{(c)}{=} & \text{ Const} + a_{ij}\theta_{ij}^2 + b_{ij}\theta_{ij} + h_{ij}(\theta_{ij}),
\end{aligned}
$$

where Const denotes the constant terms with respect to the variable of interest θ_{ij} and may vary line by line; (a) uses the definition of $\boldsymbol{r}_i = \tilde{\boldsymbol{X}}\hat{\boldsymbol{\theta}}_i$, (b) expands the squared norm and moves the constant terms into Const, and (c) is due to $v_i = \|\tilde{\boldsymbol{x}}_i\|^2$ and the definitions of a_{ij} and b_{ij}. Thus, we have shown that the line 3 of Algorithm 4.1 is given by (4.14).

In fact, this update can be expressed as the so-called proximal operator [17] for the regularizer h_{ij}, under some scaling. For a lower-semicontinuous function h, we denote by the following operator

$$
\mathcal{Q}_h(a, b) = \arg\min_{\theta} \; a\theta^2 + b\theta + h(\theta), \tag{4.31}
$$

and the proximal operator

$$
\text{prox}_h(\tilde{\theta}) = \arg\min_{\theta} \; \frac{1}{2}(\theta - \tilde{\theta})^2 + h(\theta). \tag{4.32}
$$

It is easy to verify that

$$
\mathcal{Q}_h(a, b) = \text{prox}_{\frac{1}{2a}h}\left(-\frac{b}{2a}\right). \tag{4.33}
$$

Therefore, according to (4.33), it suffices to investigate into the proximal operator

computations for the regularizers, and we will present the closed-form expressions for the proximal operators of different classes of regularizers we consider in Section 4.3.3, including L_0L_2, L_1 regularizers and relaxation regularizer g (4.11), in the following sections.

4.A.1.2 Regularizers for relaxations

The expression and the proximal operator of g in (4.11) are derived and presented in [119]. For completeness, we will summarize the results.

Interval relaxation: Recall that when $\underline{z} = 0, \bar{z} = 1$, the regularizer g becomes

$$\psi(\theta; \lambda_0, \lambda, M) = \min_{z,s} \ \lambda_0 z + \lambda_2 s, \quad \text{s.t.} \ \ sz \geq \theta^2, \ |\theta| \leq Mz, \ z \in [0, 1].$$

We summarize different regimes and cases of ψ in Table 4.3 (also in (4.8)), according to [119].

Table 4.3: Summary of different regimes and cases of ψ

Regime	Range of $\lvert\theta\rvert$	$\psi(\theta; \lambda_0, \lambda_2, M)$	z^*	s^*
$\sqrt{\lambda_0/\lambda_2} \leq M$	$[0, \sqrt{\lambda_0/\lambda_2})$	$2\sqrt{\lambda_0\lambda_2}\lvert\theta\rvert$	$\sqrt{\lambda_2/\lambda_0}\lvert\theta\rvert$	$\sqrt{\lambda_0/\lambda_2}\lvert\theta\rvert$
	$(\sqrt{\lambda_0/\lambda_2}, M]$	$\lambda_0 + \lambda_2\theta^2$	1	θ^2
	(M, ∞)	∞	$\varnothing$	$\varnothing$
$\sqrt{\lambda_0/\lambda_2} > M$	$[0, M]$	$(\lambda_0/M + \lambda_2 M)\lvert\theta\rvert$	$\lvert\theta\rvert/M$	$\lvert\theta\rvert M$
	(M, ∞)	∞	$\varnothing$	$\varnothing$

Given non-negative parameters λ and M, we define the boxed soft-thresholding operator $\mathcal{T} : \mathbb{R} \to \mathbb{R}$

$$\mathcal{T}(x; \lambda, M) := \begin{cases} 0 & \text{if } |x| \leq \lambda \\ (|x| - \lambda)\,\text{sign}(x) & \text{if } \lambda \leq |x| \leq \lambda + M \\ M\,\text{sign}(x) & \text{o.w.} \end{cases}. \tag{4.34}$$

This is the proximal operator for the boxed L_1 regularizer $h(x) = \lambda|x| + \chi\{|x| \leq M\}$.

Then, according to [119], the proximal operator of ψ is given by

$$\begin{aligned}
&\mathrm{prox}_{\psi}(\tilde{\theta};\lambda_0,\lambda_2,M)\\
&=\arg\min_{\theta}\ \frac{1}{2}(\theta-\tilde{\theta})^2+\psi(\theta;\lambda_0,\lambda_2,M)\\
&=\begin{cases}
\mathcal{T}(\tilde{\theta};2\sqrt{\lambda_0\lambda_2},M) & \text{if } |\tilde{\theta}|\le 2\sqrt{\lambda_0\lambda_2}+\sqrt{\lambda_0/\lambda_2} \text{ and } \sqrt{\lambda_0/\lambda_2}\le M\\
\mathcal{T}(\tilde{\theta}/(1+2\lambda_2);0,M) & \text{if } |\tilde{\theta}|> 2\sqrt{\lambda_0\lambda_2}+\sqrt{\lambda_0/\lambda_2} \text{ and } \sqrt{\lambda_0/\lambda_2}\le M\\
\mathcal{T}(\tilde{\theta};\lambda_0/M+\lambda_2 M,M) & \text{if } \sqrt{\lambda_0/\lambda_2}> M
\end{cases}.
\end{aligned} \tag{4.35}$$

Based on this, we define the following quadratic minimization oracle

$$\mathcal{Q}_{\psi}(a,b;\lambda_0,\lambda_2,M):=\arg\min_{x}\ ax^2+bx+\psi(x;\lambda_0,\lambda_2,M)=\mathrm{prox}_{\psi}\left(-\frac{b}{2a};\frac{\lambda_0}{2a},\frac{\lambda_2}{2a},M\right). \tag{4.36}$$

Fixed z: Recall that when $\underline{z}=\bar{z}=z\in\{0,1\}$, the regularizer g becomes φ in (4.12), i.e.

$$\begin{aligned}
\varphi(\theta;z,\lambda_0,\lambda_2,M):=&\min_{s}\quad \lambda_0 z+\lambda_2 s\\
&\text{s.t.}\quad sz\ge\theta^2,|\theta|\le Mz,z\in[0,1]\\
=&\begin{cases}
0 & \text{if } z=0 \text{ and } |\theta|=0\\
\infty & \text{if } z=0 \text{ and } |\theta|>0\\
\lambda_0+\lambda_2\theta^2 & \text{if } z=1 \text{ and } |\theta|\le M\\
\infty & \text{if } z=1 \text{ and } |\theta|> M
\end{cases},
\end{aligned} \tag{4.37}$$

and its corresponding proximal operator is

$$\begin{aligned}
\mathrm{prox}_{\varphi}(\tilde{\theta};z,\lambda_0,\lambda_2,M)&=\arg\min_{\theta}\ \frac{1}{2}(\theta-\tilde{\theta})^2+\varphi(\theta;z,\lambda_0,\lambda_2,M)\\
&=\begin{cases}
0 & \text{if } z=0\\
\mathcal{T}(\tilde{\theta}/(1+2\lambda_2);0,M) & \text{if } z=1
\end{cases}.
\end{aligned} \tag{4.38}$$

Based on this, we define the following regularized quadratic minimization oracle

$$\begin{aligned}\mathcal{Q}_\varphi(a,b;z,\lambda_0,\lambda_2,M) &:= \arg\min_x \; ax^2+bx+\varphi(x;z,\lambda_0,\lambda_2,M)\\ &= \mathrm{prox}_\varphi\left(-\frac{b}{2a};z,\frac{\lambda_0}{2a},\frac{\lambda_2}{2a},M\right)\end{aligned} \tag{4.39}$$

4.A.1.3 $\ell_0\ell_2$ regularizers

We derive the closed-form expression for the proximal operator for the L_0L_2 regularizer

$$h(\theta)=\lambda_0\mathbf{1}\{\theta\neq 0\}+\lambda_2\theta^2+\chi\{|\theta|\le M\},$$

where $M>0$ could be ∞.

The proximal operator of h is

$$\mathrm{prox}_h(\tilde\theta;\lambda_0,\lambda_2,M)=\arg\min_{|\theta|\le M} q(\theta):=\frac{1}{2}(\theta-\tilde\theta)^2+\lambda_0\mathbf{1}\{\theta\neq 0\}+\lambda_2\theta^2.$$

When $\theta=0$, we have $q(0)=\frac{1}{2}\tilde\theta^2$; when $\theta\neq 0$, we have $q(\theta)=\lambda_0+\lambda_2\theta^2+\frac{1}{2}(\theta-\tilde\theta)^2$, which is minimized at $\theta'=\min\{|\tilde\theta|/(1+2\lambda_2),M\}\,\mathrm{sign}(\tilde\theta)$.

Without loss of generality, we assume $\tilde\theta>0$. If $\frac{\tilde\theta}{1+2\lambda_2}>M$, then $\theta'=M$, and

$$q(\theta')=\lambda_0+\lambda_2M^2+\frac{1}{2}(M-\tilde\theta)^2.$$

The root of $q(\theta')=q(0)$ is $\tilde\theta=(\frac{1}{2}+\lambda_2)M+\frac{\lambda_0}{M}$.

If, on the other hand, $\frac{\tilde\theta}{1+2\lambda_2}\le M$, then $\theta'=\frac{\tilde\theta}{1+2\lambda_2}$, and

$$q(\theta')=\lambda_0+\frac{\lambda_2\tilde\theta^2}{1+2\lambda_2}.$$

The root of $q(\theta')=q(0)$ is $\tilde\theta=\sqrt{2\lambda_0(1+2\lambda_2)}$.

Therefore, we obtain the following closed-form expression for the L_0L_2 regularizer

$$\begin{aligned}
&\mathrm{prox}_h(\tilde{\theta};\lambda_0,\lambda_2,M) = \arg\min_{|\theta|\le M} q(\theta) \\
&= \begin{cases}
\{M\,\mathrm{sign}(\tilde{\theta})\}, & \text{if } |\tilde{\theta}| > \max\left\{(\frac{1}{2}+\lambda_2)M + \frac{\lambda_0}{M}, (1+2\lambda_2)M\right\} \\
\{0, M\,\mathrm{sign}(\tilde{\theta})\}, & \text{if } |\tilde{\theta}| = (\frac{1}{2}+\lambda_2)M + \frac{\lambda_0}{M} > (1+2\lambda_2)M \\
\{0\}, & \text{if } |\tilde{\theta}| \in \left((1+\lambda_2)M, (\frac{1}{2}+\lambda_2)M + \frac{\lambda_0}{M}\right) \\
\{\frac{\tilde{\theta}}{1+2\lambda_2}\}, & \text{if } |\tilde{\theta}| \in \left(\sqrt{2\lambda_0(1+2\lambda_2)}, (1+2\lambda_2)M\right] \\
\{0, \frac{\tilde{\theta}}{1+2\lambda_2}\}, & \text{if } |\tilde{\theta}| = \sqrt{2\lambda_0(1+2\lambda_2)} \le (1+2\lambda_2)M \\
\{0\}, & \text{if } |\tilde{\theta}| < \min\{\sqrt{2\lambda_0(1+2\lambda_2)}, (1+2\lambda_2)M\}
\end{cases}
\end{aligned} \tag{4.40}$$

Note that when $M = \infty$, (4.40) (the last three conditions) recovers the closed-form expression for $\ell_0\ell_2$ regularizer provided in [116].

4.A.1.4 Diagonal update

We show that the update of $\hat{\theta}_{ii}$ in the line 6 of Algorithm 4.1 is given by (4.15). For any i, we have

$$\begin{aligned}
F(\hat{\boldsymbol{\Theta}} - \hat{\theta}_{ii}\boldsymbol{E}_{ii} + \theta_{ii}\boldsymbol{E}_{ii}) &= \mathrm{Const} - \log\theta_{ii} + \frac{1}{\theta_{ii}}\|\tilde{\boldsymbol{X}}\hat{\boldsymbol{\theta}}_i - \hat{\theta}_{ii}\tilde{\boldsymbol{x}}_i + \theta_{ii}\tilde{\boldsymbol{x}}_i\|^2 \\
&\overset{(a)}{=} \mathrm{Const} - \log\theta_{ii} + \frac{1}{\theta_{ii}}\|\boldsymbol{r}_i - \hat{\theta}_{ii}\tilde{\boldsymbol{x}}_i + \theta_{ii}\tilde{\boldsymbol{x}}_i\|^2 \\
&\overset{(b)}{=} \mathrm{Const} - \log\theta_{ii} + \frac{1}{\theta_{ii}}(\|\boldsymbol{e}_i\|^2 + 2\theta_{ii}\boldsymbol{e}_i^\top\tilde{\boldsymbol{x}}_i + \theta_{ii}^2\|\tilde{\boldsymbol{x}}_i\|^2) \\
&\overset{(c)}{=} \mathrm{Const} - \log\theta_{ii} + \frac{\|\boldsymbol{e}_i\|^2}{\theta_{ii}} + \theta_{ii}v_i,
\end{aligned}$$

where Const denotes the constant terms with respect to the variable of interest θ_{ii} and may vary line by line; (a) and (b) uses the definitions of $\boldsymbol{r}_i = \tilde{\boldsymbol{X}}\hat{\boldsymbol{\theta}}_i$ and $\boldsymbol{e}_i = \boldsymbol{r}_i - \hat{\theta}_{ii}\tilde{\boldsymbol{x}}_i$, and (c) is due to $v_i = \|\tilde{\boldsymbol{x}}_i\|^2$ and $2\boldsymbol{e}_i^\top\tilde{\boldsymbol{x}}_i$ absorbed into Const.

Since the function is convex in θ_{ii}, by computing the first-order condition and taking the positive root, we get

$$\arg\min_{\theta_{ii}} F(\hat{\boldsymbol{\Theta}} - \hat{\theta}_{ii}\boldsymbol{E}_{ii} + \theta_{ii}\boldsymbol{E}_{ii}) = \arg\min_{\theta} -\log\theta + \theta v_i + \frac{\|\boldsymbol{e}_i\|^2}{\theta} = \frac{1+\sqrt{1+4v_i\|\boldsymbol{e}_i\|^2}}{2v_i}.$$

4.A.2 Convergence guarantee of Algorithm 4.1

In this section, we consider a more general convergence statement about the unified formulation (4.13), with the following assumption on h_{ij}:

Assumption 4.3. *Assume that for each $1 \le i < j \le p$, $h_{ij}(\theta)$ is convex in θ. In addition, there exist two constants $c_1, c_2 \ge 0$ but $c_1 + c_2 > 0$, such that for any $1 \le i < j \le p$, we have*

$$h_{ij}(\theta) \ge \min\{c_1|\theta|, c_2\theta^2\}.$$

It is easy to see that the usual ℓ_1, ℓ_2 penalties and their combinations satisfy Assumption 4.3. The following proposition states that the relaxation regularizer g also satisfies this assumption.

Proposition 4.2. *For any $\underline{z}_{ij} \le \bar{z}_{ij} \in \{0, 1\}$, $g(\theta; \lambda_0, \lambda_2, M, \underline{z}_{ij}, \bar{z}_{ij})$ satisfies Assumption 4.3 with $c_1 = 2\sqrt{\lambda_0\lambda_2}, c_2 = 0$.*

Proof. Based on the definition of φ and ψ in different cases, using the inequality $a^2+b^2 \ge 2ab$, it is easy to see that $\psi(\theta; \lambda_0, \lambda_2, M) \ge 2\sqrt{\lambda_0\lambda_2}|\theta|$ and $\varphi(\theta; z, \lambda_0, \lambda, M) \ge 2\sqrt{\lambda_0\lambda_2}|\theta|$. □

Lemma 4.1. *Under Assumption 4.3, given any $U \ge F^* = \min_{\boldsymbol{\Theta} \in \mathbb{S}^p} F(\boldsymbol{\Theta})$, there exist constants $u_\theta \ge l_\theta > 0$ and $u_r > 0$, for any $\boldsymbol{\Theta}$ such that $F(\boldsymbol{\Theta}) \le U$ and any $i \in [p]$, we have*

$$l_\theta \le \theta_{ii} \le u_\theta \quad \textit{and} \quad \|\tilde{\boldsymbol{X}}\boldsymbol{\theta}_i\| \le u_r.$$

Proof. For any $\boldsymbol{\Theta}$ such that $F(\boldsymbol{\Theta}) \le U$, let $k = \arg\max_{i \in [p]} \theta_{ii}$. Then, we have

$$\begin{aligned} U &\ge F(\boldsymbol{\Theta}) \\ &= \sum_{i=1}^{p} \left(-\log\theta_{ii} + \frac{1}{\theta_{ii}} \|\tilde{\boldsymbol{X}}\boldsymbol{\theta}_i\|^2 \right) + \sum_{i<j} h_{ij}(\theta_{ij}) \\ &\ge -p\log\theta_{kk} + \frac{1}{\theta_{kk}} \|\tilde{\boldsymbol{X}}\boldsymbol{\theta}_k\|^2 + \sum_{i \ne k} \min\{c_1|\theta_{ik}|, c_2\theta_{ik}^2\}, \end{aligned} \tag{4.41}$$

where the last line is because (i) $-\log\theta_{ii} \geq -\log\theta_{kk}$ by definition of k; (ii) by Assumption 4.3, $h_{ij}(\theta_{ij}) \geq \min\{c_1|\theta_{ij}|, c_2\theta_{ij}^2\} \geq 0$; (iii) $\frac{1}{\theta_{ii}}\|\tilde{\boldsymbol{X}}\boldsymbol{\theta}_i\|^2$ is nonnegative for any $i \neq k$.

Now define $\boldsymbol{\beta}_k \in \mathbb{R}^p$, with $\beta_{kk} = 0$ and $\beta_{ki} = -\theta_{ik}/\theta_{kk}$, then we can rewrite (4.41) into

$$\begin{aligned}
U &\geq -p\log\theta_{kk} + \frac{1}{\theta_{kk}}\|\theta_{kk}\tilde{\boldsymbol{x}}_k - \tilde{\boldsymbol{X}}\theta_{kk}\boldsymbol{\beta}_k\|^2 + \sum_{i\neq k}\min\{c_1|\theta_{ik}|, c_2\theta_{ik}^2\} \\
&= -p\log\theta_{kk} + \theta_{kk}\|\tilde{\boldsymbol{x}}_k - \tilde{\boldsymbol{X}}\boldsymbol{\beta}_k\|^2 + \sum_{i\neq k}\min\{c_1\theta_{kk}|\beta_{ki}|, c_2\theta_{kk}^2\beta_{ki}^2\} \\
&\overset{(a)}{\geq} -p\log\theta_{kk} + \theta_{kk}\max\left\{\frac{1}{2}\|\tilde{\boldsymbol{x}}_k\|^2 - \|\tilde{\boldsymbol{X}}\boldsymbol{\beta}_k\|^2, 0\right\} + \sum_{i\neq k}\min\{c_1\theta_{kk}|\beta_{ki}|, c_2\theta_{kk}^2\beta_{ki}^2\} \\
&\overset{(b)}{\geq} -p\log\theta_{kk} + \theta_{kk}\max\left\{\frac{1}{2}s_{\min} - L_{\tilde{\boldsymbol{X}}}^2\|\boldsymbol{\beta}_k\|^2, 0\right\} + \sum_{i\neq k}\min\{c_1\theta_{kk}|\beta_{ki}|, c_2\theta_{kk}^2\beta_{ki}^2\},
\end{aligned} \tag{4.42}$$

where (a) uses the fact that $\|\boldsymbol{a}+\boldsymbol{b}\|^2 \leq 2(\|\boldsymbol{a}\|^2 + \|\boldsymbol{b}\|^2)$ with $\boldsymbol{a} = \tilde{\boldsymbol{x}}_k - \tilde{\boldsymbol{X}}\boldsymbol{\beta}_k$ and $\boldsymbol{b} = \tilde{\boldsymbol{X}}\boldsymbol{\beta}_k$; in (b), we define $s_{\min} = \min_i v_i = \min_i \|\tilde{\boldsymbol{x}}_i\|^2 > 0$ and $L_{\tilde{\boldsymbol{X}}} = \|\tilde{\boldsymbol{X}}\|$.

Now if $\|\boldsymbol{\beta}_k\| \leq \epsilon := L_{\tilde{\boldsymbol{X}}}\sqrt{s_{\min}}/2$, then it follows from (4.42) that $-p\log\theta_{kk} + \frac{1}{4}s_{\min}\theta_{kk} \leq U$, from which we can deduce there exists $u_1 > 0$, such that $\theta_{kk} \leq u_1$. On the other hand, if $\|\boldsymbol{\beta}_k\| > \epsilon$, then there exists a j such that $|\beta_{kj}| \geq \epsilon/\sqrt{p}$, again by (4.42), we have $-p\log\theta_{kk} + \min\{c_1\theta_{kk}\epsilon/\sqrt{p}, c_2\theta_{kk}\epsilon^2/p\} \leq U$, and thus there exists $u_2 > 0$, such that $\theta_{kk} \leq u_2$. Therefore, $\theta_{kk} = \max_i\theta_{ii} \leq \max\{u_1, u_2\}$, and by taking $u_\theta = \max\{u_1, u_2\}$, we get the upper bound.

As for the lower bound on θ_{ii}, let $\ell = \arg\min_i\theta_{ii}$, by nonnegativity of $\frac{1}{\theta_{ii}}\|\tilde{\boldsymbol{X}}\boldsymbol{\theta}_i\|^2$ and h_{ij}'s, we have

$$U \geq -\sum_{i=1}^{p}\log\theta_{ii} = -\log\theta_{\ell\ell} - \sum_{i\neq\ell}\log\theta_{ii} \geq -\log\theta_{\ell\ell} - (p-1)\log u_\theta.$$

Therefore, $\theta_{\ell\ell} = \min_i\theta_{ii} \geq \exp(-M - (p-1)\log u_\theta)$, and we obtain the lower bound $l_\theta = \exp(-M - (p-1)\log u_\theta)$.

Again by nonnegativity of $\frac{1}{\theta_{ii}}\|\tilde{\boldsymbol{X}}\boldsymbol{\theta}_i\|^2$ and h_{ij}'s, we have for any j,

$$U \geq -\sum_{i=1}^{p} \log\theta_{ii} + \frac{1}{\theta_{jj}}\|\tilde{\boldsymbol{X}}\boldsymbol{\theta}_j\|^2 = -p\log u_\theta + \frac{1}{u_\theta}\|\tilde{\boldsymbol{X}}\boldsymbol{\theta}_j\|^2.$$

Therefore, $\|\tilde{\boldsymbol{X}}\boldsymbol{\theta}_j\|^2 \leq u_\theta(U + p\log u_\theta)$, and we obtain the upper bound $u_r = \sqrt{u_\theta(U + p\log u_\theta)}$. □

Corollary 4.1. *Let* $f(\boldsymbol{\Theta}) = \sum_{i=1}^{p}\left(-\log\theta_{ii} + \frac{1}{\theta_{ii}}\|\tilde{\boldsymbol{X}}\boldsymbol{\theta}_i\|^2\right)$. *Under Assumption 4.3, given any* $F^{(0)} \geq F^* = \min_{\boldsymbol{\Theta}\in\mathbb{S}^p} F(\boldsymbol{\Theta})$, *there exist constants* L, μ_{ij}, L_{ij} *such that over* $\{\boldsymbol{\Theta} : F(\boldsymbol{\Theta}) \leq F^{(0)}\}$, *the objective function* $F(\boldsymbol{\Theta})$ *in* (4.13) *satisfies*

(a) ∇f *is L-Lipschitz*

(b) ∇f *is* $\{L_{ij}\}$*-coordinatewise Lipschitz*

(c) F *is* $\{\mu_{ij}\}$*-coordinatewise strongly convex*

Proof. We will show (b) and (c), and (a) follows from (b).

From the derivation of the off-diagonal update in Section 4.A.1.1, we can easily see that the second derivative of f with respect to the off-diagonal entry θ_{ij} (for any $i < j$) is given by

$$\nabla^2_{\theta_{ij}} f(\theta_{ij}) = \frac{v_i}{\theta_{jj}} + \frac{v_j}{\theta_{ii}}. \tag{4.43}$$

From the derivation of the on-diagonal update in Section 4.A.1.4, we see that the second derivative of f with respect to the on-diagonal entry θ_{ii} (for any i) is given by

$$\nabla^2_{\theta_{ii}} f(\theta_{ii}) = \frac{1}{\theta_{ii}^2} + \frac{\|\tilde{\boldsymbol{X}}_{-i}\boldsymbol{\theta}_{i,-i}\|^2}{2\theta_{ii}^2}, \tag{4.44}$$

where $\tilde{\boldsymbol{X}}_{-i} \in \mathbb{R}^{n\times(p-1)}$ is the data matrix without i-th column, and $\boldsymbol{\theta}_{i,-i} \in \mathbb{R}^{p-1}$ is the vector of $\boldsymbol{\theta}_i$ without i-th component.

(b) By Lemma 4.1, we have $\theta_{ii} \geq l_\theta$, and it follows from (4.43) that $\nabla^2_{\theta_{ij}} f(\theta_{ij}) \leq (v_i + v_j)/l_\theta$. Therefore, we have ∇f is L_{ij}-Lipschitz with respect to θ_{ij}, where $L_{ij} = (v_i + v_j)/l_\theta$.

Again by Lemma 4.1, we have $\theta_{ii} \geq l_\theta$ and $\|\tilde{\boldsymbol{X}}\boldsymbol{\theta}_i\|^2 \leq u_r^2$, and thus

$$\nabla f^2_{\theta_{ii}}(\theta_{ii}) = \frac{1}{\theta_{ii}^2} + \frac{\|\tilde{\boldsymbol{X}}\boldsymbol{\theta}_i - \theta_{ii}\tilde{\boldsymbol{x}}_i\|^2}{2\theta_{ii}^2} \leq \frac{1 + \|\tilde{\boldsymbol{X}}\boldsymbol{\theta}_i\|^2 + \theta_{ii}^2\|\tilde{\boldsymbol{x}}_i\|^2}{\theta_{ii}^2} \leq \frac{1+u_r^2}{l_\theta^2} + v_i.$$

Therefore, we have ∇f is L_{ii}-Lipschitz with respect to θ_{ii}, where $L_{ii} = (1+u_r^2)/l_\theta^2 + v_i$.

(c) By Lemma 4.1, we have $\theta_{ii} \leq u_\theta$, so $\nabla^2_{\theta_{ij}} f(\theta_{ij}) \geq (v_i + v_j)/u_\theta$. Therefore, we have ∇f is μ_{ij}-strongly convex with respect to θ_{ij}, where $\mu_{ij} = (v_i + v_j)/u_\theta$.

Similarly, we have ∇f is μ_{ii}-strongly convex with respect to θ_{ii} with $\mu_{ii} = 1/u_\theta^2$. □

Theorem 4.6. *Under Assumption 4.3, given any initialization $\boldsymbol{\Theta}^{(0)}$, let $\boldsymbol{\Theta}^{(t)}$ be the t-th iterate generated by Algorithm 4.1, then there exists a constant C that depends on $\boldsymbol{\Theta}^{(0)}$, such that for any $t \geq 1$,*

$$F(\boldsymbol{\Theta}^{(t)}) - F^* \leq \frac{C}{t},$$

where $F^* = \min_{\boldsymbol{\Theta} \in \mathbb{S}^p} F(\boldsymbol{\Theta})$.

Proof. It is not hard to see that Algorithm 4.1 is a descent algorithm, meaning that the objective function decreases after each coordinate update. Therefore, we must have

$$F(\boldsymbol{\Theta}^{(t)}) \leq F(\boldsymbol{\Theta}^{(0)}),$$

i.e. $\boldsymbol{\Theta}^{(t)} \in \{\boldsymbol{\Theta} : F(\boldsymbol{\Theta}) \leq F(\boldsymbol{\Theta}^{(0)})\}$.

Since Assumption 4.3 holds, invoking Corollary 4.1 with $F^{(0)} = F(\boldsymbol{\Theta}^{(0)})$, we get f is coordinatewise-Lipschitz and F is coordinatewise-strong convex with some parameters depending on $F^{(0)}$, and thus on $\boldsymbol{\Theta}^{(0)}$. According to [123], we get the sublinear rate of convergence of Algorithm 4.1, i.e.

$$F(\boldsymbol{\Theta}^{(t)}) - F^* \leq \frac{C}{t},$$

where the constant C depends on $\boldsymbol{\Theta}^{(0)}$. □

Remark 4.5. *According to Proposition 4.2, the regularizers $g(\theta_{ij}; \lambda_0, \lambda_2, M, \underline{z}_{ij}, \bar{z}_{ij})$ in F_{node} satisfy Assumption 4.3, and thus Theorem 4.6 applies to F_{node}, which is exactly*

Theorem 4.1 in the main text. Furthermore, as we mentioned earlier, $h_{ij}(\theta_{ij}) = \lambda_1|\theta_{ij}|$ also satisfies Assumption 4.3, and thus Theorem 4.6 also provides convergence guarantee for the convex reformulation of symmetric lasso formulation.

4.A.3 Dual bound

In this section, we first provide the proof for Theorem 4.2 in Section 4.A.3.1, deriving the Lagrangian dual of the node relaxation objective F_{node}. We then summarize how to compute the convex conjugate of ψ and φ as two cases of g in Section 4.A.3.2. Finally, we provide the proof for Proposition 4.1 in Section 4.A.3.3.

4.A.3.1 Proof of Theorem 4.2

Proof of Theorem 4.2. We introduce auxiliary primal variables $\boldsymbol{r}_i = \tilde{\boldsymbol{X}}\boldsymbol{\theta}_i$ to rewrite the problem into the following formulation

$$\min_{\boldsymbol{\Theta}\in\mathbb{S}^p} \sum_{i=1}^{p}(-\log\theta_{ii} + \frac{1}{\theta_{ii}}\|\boldsymbol{r}_i\|^2) + \sum_{i<j} g(\theta_{ij};\lambda_0,\lambda_2,M,\underline{z}_{ij},\bar{z}_{ij}), \quad \text{s.t.} \quad \boldsymbol{r}_i = \tilde{\boldsymbol{X}}\boldsymbol{\theta}_i,\ \forall i\in[p] \tag{4.45}$$

By dualizing the constraints in (4.45), we can write the Lagrangian as

$$\mathcal{L}(\boldsymbol{\Theta},\boldsymbol{r};\boldsymbol{\nu}) = \sum_{i=1}^{p}(-\log\theta_{ii} + \frac{1}{\theta_{ii}}\|\boldsymbol{r}_i\|^2 + \langle\boldsymbol{\nu}_i, \boldsymbol{r}_i - \tilde{\boldsymbol{X}}\boldsymbol{\theta}_i\rangle) + \sum_{i<j} g(\theta_{ij};\lambda_0,\lambda_2,M,\underline{z}_{ij},\bar{z}_{ij}). \tag{4.46}$$

The Lagrangian dual is given by $D(\boldsymbol{\nu}) = \min_{\boldsymbol{\Theta}\in\mathbb{S}^p}\mathcal{L}(\boldsymbol{\Theta},\boldsymbol{r};\boldsymbol{\nu})$. Since the Slater's condition holds [27], we have the strong duality holds, i.e.

$$\min_{\boldsymbol{\Theta}\in\mathbb{S}^p} F_{\text{node}}(\boldsymbol{\Theta}) = \max_{\boldsymbol{\nu}} D(\boldsymbol{\nu}).$$

Minimizing (4.46) with respect to $\boldsymbol{r}_i$, we get

$$\boldsymbol{r}_i = -\frac{\theta_{ii}}{2}\boldsymbol{\nu}_i. \tag{4.47}$$

Plugging this back to the Lagrangian (4.46), we get

$$\theta_{ii} = \arg\min_{\theta} \ -\log\theta + \theta(-\|\boldsymbol{\nu}_i\|^2/4 - \tilde{\boldsymbol{x}}_i^\top \boldsymbol{\nu}_i),$$

which yields

$$\theta_{ii} = \frac{1}{-\|\boldsymbol{\nu}_i\|^2/4 - \tilde{\boldsymbol{x}}_i^\top \boldsymbol{\nu}_i} \quad \text{if} \quad -\|\boldsymbol{\nu}_i\|^2/4 - \tilde{\boldsymbol{x}}_i^\top \boldsymbol{\nu}_i > 0. \tag{4.48}$$

If $-\|\boldsymbol{\nu}_i\|^2/4 - \tilde{\boldsymbol{x}}_i^\top \boldsymbol{\nu}_i \leq 0$, then $\theta_{ii} \to \infty$, and the minimum value is $-\infty$, which cannot be achieved.

As for $\theta_{ij} = \theta_{ji}$,

$$\begin{aligned}
\theta_{ij} = \theta_{ji} &= \arg\min_{\theta} \ (-\tilde{\boldsymbol{x}}_j^\top \boldsymbol{\nu}_i - \tilde{\boldsymbol{x}}_i^\top \boldsymbol{\nu}_j)\theta + g(\theta; \lambda_0, \lambda_2, M, \underline{z}_{ij}, \bar{z}_{ij}) \\
&= \arg\max_{\theta} \ (\tilde{\boldsymbol{x}}_j^\top \boldsymbol{\nu}_i + \tilde{\boldsymbol{x}}_i^\top \boldsymbol{\nu}_j)\theta - g(\theta; \lambda_0, \lambda_2, M, \underline{z}_{ij}, \bar{z}_{ij}) \\
&\in \partial g^*(\tilde{\boldsymbol{x}}_j^\top \boldsymbol{\nu}_i + \tilde{\boldsymbol{x}}_i^\top \boldsymbol{\nu}_j; \lambda_0, \lambda_2, M, \underline{z}_{ij}, \bar{z}_{ij}).
\end{aligned} \tag{4.49}$$

Therefore, plugging (4.47), (4.48) and (4.49) into the Lagrangian function (4.46), we get the Lagrangian dual problem:

$$\max_{\boldsymbol{\nu}} \ D(\boldsymbol{\nu}) = p + \sum_{i=1}^{p} \log(-\|\boldsymbol{\nu}_i\|^2/4 - \tilde{\boldsymbol{x}}_i^\top \boldsymbol{\nu}_i) - \sum_{i<j} g^*(\tilde{\boldsymbol{x}}_j^\top \boldsymbol{\nu}_i + \tilde{\boldsymbol{x}}_i^\top \boldsymbol{\nu}_j; \lambda_0, \lambda_2, M, \underline{z}_{ij}, \bar{z}_{ij}). \tag{4.50}$$

□

4.A.3.2 Computing the convex conjugates

Convex Conjugate of ψ: We consider the Fenchel conjugate ψ^* of ψ:

$$\psi^*(\alpha; \lambda_0, \lambda_2, M) := \sup_{\theta} \ \alpha\theta - \psi(\theta; \lambda_0, \lambda_2, M). \tag{4.51}$$

According to [119], when $\sqrt{\lambda_0/\lambda_2} \leq M$,

$$\psi^*(\alpha; \lambda_0, \lambda_2, M) = \min_{\gamma} \left[\frac{(\gamma - \alpha)^2}{4\lambda_2} - \lambda_0 \right]_+ + M|\gamma|; \tag{4.52}$$

when $\sqrt{\lambda_0/\lambda_2} > M$,

$$\psi^*(\alpha; \lambda_0, \lambda_2, M) = \min_{\mu} \ M|\mu| \quad \text{s.t.} \quad |\alpha| - \mu \leq \lambda_0/M + \lambda_2 M. \tag{4.53}$$

We summarize different regimes and cases of ψ^* as follows

Table 4.4: Summary of different regimes and cases of ψ^*

Regime	Range of $\lvert\alpha\rvert$	$\psi^*(\alpha; \lambda_0, \lambda_2, M)$	$\theta^* \in \partial\psi^*(\alpha)$	γ^*/μ^*
$\sqrt{\lambda_0/\lambda_2} \leq M$	$[0, 2\sqrt{\lambda_0\lambda_2})$	0	0	0
	$(2\sqrt{\lambda_0\lambda_2}, 2\lambda_2 M]$	$\frac{\alpha^2}{4\lambda_2} - \lambda_0$	$\frac{\alpha}{2\lambda_2}$	0
	$(2\lambda_2 M, \infty)$	$M\lvert\alpha\rvert - (\lambda_0 + \lambda_2 M^2)$	$M\,\mathrm{sign}(\alpha)$	$\alpha - 2M\lambda_2\,\mathrm{sign}(\alpha)$
$\sqrt{\lambda_0/\lambda_2} > M$	$[0, \lambda_0/M + \lambda_2 M]$	0	0	0
	$(\lambda_0/M + \lambda_2 M, \infty)$	$M\lvert\alpha\rvert - (\lambda_0 + \lambda_2 M^2)$	$M\,\mathrm{sign}(\alpha)$	$\lvert\alpha\rvert - (\lambda_0/M + \lambda_2 M)$

Convex conjugate of φ: We consider the Fenchel conjugate φ^* of φ:

$$\varphi^*(\alpha; z, \lambda_0, \lambda_2, M) := \sup_{\theta} \ \alpha\theta - \varphi(\theta; z, \lambda_0, \lambda_2, M). \tag{4.54}$$

We summarize different regimes and cases of φ^* as follows

Table 4.5: Summary of different regimes and cases of φ^*

Regime	Range of $\lvert\alpha\rvert$	$\varphi^*(\alpha; z, \lambda_0, \lambda_2, M)$	$\theta^* \in \partial\varphi^*(\alpha; z)$
$z = 0$	$[0, \infty)$	0	0
$z = 1, \lambda_2 > 0$	$[0, 2\lambda_2 M]$	$\frac{\alpha^2}{4\lambda_2} - \lambda_0$	$\frac{\alpha}{2\lambda_2}$
	$(2\lambda_2 M, \infty)$	$M\lvert\alpha\rvert - (\lambda_0 + \lambda_2 M^2)$	$M\,\mathrm{sign}(\alpha)$
$z = 1, \lambda_2 = 0$	$[0, \infty)$	$M\lvert\alpha\rvert - \lambda_0$	$M\,\mathrm{sign}(\alpha)$

4.A.3.3 Proof of Proposition 4.1

To prove Proposition 4.1, we start with the following proposition:

Proposition 4.3. *Denote by*

$$c(\lambda_0, \lambda_2, M) = \begin{cases} 2\sqrt{\lambda_0\lambda_2} & \text{if } \sqrt{\lambda_0/\lambda_2} \leq M \\ \lambda_0/M + \lambda_2 M & \text{o.w.} \end{cases}. \tag{4.55}$$

The following statements hold

(a) $\text{prox}_{\psi}(\tilde{\beta}; \lambda_0, \lambda_2, M) = 0 \iff |\tilde{\beta}| \leq c(\lambda_0, \lambda_2, M)$

(b) $\mathcal{Q}_{\psi}(a, b; \lambda_0, \lambda_2, M) = 0 \iff |b| \leq c(\lambda_0, \lambda_2, M)$

(c) $\psi^*(\alpha; \lambda_0, \lambda_2, M) = 0 \iff |\alpha| \leq c(\lambda_0, \lambda_2, M)$

The proposition can be easily verified by using (4.35), (4.36) and Table 4.4.

Proof of Proposition 4.1. Recall the definitions of $\hat{\mathcal{S}}$ and $\mathcal{F}_1$:

$$\hat{\mathcal{S}} = \{(i,j) : i < j, \hat{\theta}_{ij} \neq 0\}, \quad \text{and} \quad \mathcal{F}_1 = \{(i,j) : \underline{z}_{ij} = \bar{z}_{ij} = 1\}.$$

Additionally, we define

$$\mathcal{F}_0 = \{(i,j) : \underline{z}_{ij} = \bar{z}_{ij} = 0\}, \quad \text{and} \quad \mathcal{R} = \{(i,j) : \underline{z}_{ij} = 0, \bar{z}_{ij} = 1\}.$$

Throughout the proof, we will denote by $g^*_{ij} := g^*(\tilde{\boldsymbol{x}}_i^\top \hat{\boldsymbol{\nu}}_j + \tilde{\boldsymbol{x}}_j^\top \hat{\boldsymbol{\nu}}_i; \lambda_0, \lambda_2, M, \underline{z}_{ij}, \bar{z}_{ij})$.

Claim: We claim that for any $\hat{\theta}_{ij} = 0$, i.e. $(i,j) \in \hat{\mathcal{S}}^c$,

(a) if $(i,j) \in \mathcal{R} \cup \mathcal{F}_0$, then

$$g^*_{ij} = 0;$$

(b) if $(i,j) \in \mathcal{F}_1$, then

$$g^*_{ij} = -\lambda_0.$$

We can decompose the sum over all pairs of (i,j) into four parts:

$$\sum_{i,j} g_{ij}^* = \sum_{(i,j)\in\hat{\mathcal{S}}} g_{ij}^* + \sum_{(i,j)\in\hat{\mathcal{S}}^c\cap\mathcal{F}_1} g_{ij}^* + \sum_{(i,j)\in\hat{\mathcal{S}}^c\cap\mathcal{F}_0} g_{ij}^* + \sum_{(i,j)\in\hat{\mathcal{S}}^c\cap\mathcal{R}} g_{ij}^*.$$

With the claim, we have the last two term are 0 and the second term becomes $-\lambda_0|\mathcal{F}_1\backslash\hat{\mathcal{S}}|$. Thus, we prove the desired result.

Proof of the claim: We first note that when Algorithm 4.2 terminates, then $\mathcal{V}$ must be empty, i.e. for any $\hat{\theta}_{ij}=0$, we must have

$$0 \in \arg\min_{\theta_{ij}} F(\hat{\boldsymbol{\Theta}} - \hat{\theta}_{ij}\boldsymbol{E}_{ij} - \hat{\theta}_{ij}\boldsymbol{E}_{ji} + \theta_{ij}\boldsymbol{E}_{ij} + \theta_{ij}\boldsymbol{E}_{ji}),$$

or equivalently (according to Section 4.A.1.1),

$$\mathcal{Q}_g\left(a_{ij}, b_{ij}; \lambda_0, \lambda_2, M, \underline{z}_{ij}, \bar{z}_{ij}\right) = 0, \tag{4.56}$$

where

$$a_{ij} = \frac{v_j}{\hat{\theta}_{ii}} + \frac{v_i}{\hat{\theta}_{jj}},$$

and

$$b_{ij} = \frac{2\tilde{\boldsymbol{x}}_j^\top(\hat{\boldsymbol{r}}_i - \hat{\theta}_{ij}\tilde{\boldsymbol{x}}_j)}{\hat{\theta}_{ii}} + \frac{2\tilde{\boldsymbol{x}}_i^\top(\hat{\boldsymbol{r}}_j - \hat{\theta}_{ij}\tilde{\boldsymbol{x}}_i)}{\hat{\theta}_{jj}} = -(\tilde{\boldsymbol{x}}_i^\top\hat{\boldsymbol{\nu}}_j + \tilde{\boldsymbol{x}}_j^\top\hat{\boldsymbol{\nu}}_j). \tag{4.57}$$

Here, (4.57) follows from $\hat{\theta}_{ij}=0$ and the dual solution definition (4.18).

Proof of (a): If $(i,j)\in\mathcal{F}_0$, then reading Table 4.5, we get $g_{ij}^*=0$.

If $(i,j)\in\mathcal{R}$, we have $g=\psi$. According to Proposition 4.3 (b), (4.56) with (4.57) implies $|b_{ij}| = |\tilde{\boldsymbol{x}}_i^\top\hat{\boldsymbol{\nu}}_j + \tilde{\boldsymbol{x}}_j^\top\hat{\boldsymbol{\nu}}_j| \le c(\lambda_0,\lambda_2,M)$, which, by Proposition 4.3 (c), implies $g_{ij}^* = \psi^*(\tilde{\boldsymbol{x}}_i^\top\hat{\boldsymbol{\nu}}_j + \tilde{\boldsymbol{x}}_j^\top\hat{\boldsymbol{\nu}}_j) = 0$.

Proof of (b): If $(i,j)\in\mathcal{F}_1$, then $g(\theta_{ij}) = \psi(\theta_{ij}; z, \lambda_0, \lambda_2, M)$ with $z=1$. According to (4.39) and (4.38) in the case of $z=1$, we have (4.56) with (4.57) implies $|b_{ij}| = |\tilde{\boldsymbol{x}}_i^\top\hat{\boldsymbol{\nu}}_j + \tilde{\boldsymbol{x}}_j^\top\hat{\boldsymbol{\nu}}_j| = 0$, which implies $g_{ij}^* = \varphi^*(\tilde{\boldsymbol{x}}_i^\top\hat{\boldsymbol{\nu}}_j + \tilde{\boldsymbol{x}}_j^\top\hat{\boldsymbol{\nu}}_j; z=1) = -\lambda_0$, according to Table 4.5. □

4.B Proofs from Section 4.4

Before proceeding with the proof of main results, we present a formal version of our discussion at the beginning of Section 4.2 that we use throughout the proofs.

Lemma 4.2. *For $j \in [p]$, let $\boldsymbol{\varepsilon}_j \in \mathbb{R}^n$ be such that*

$$\boldsymbol{\varepsilon}_j = \boldsymbol{x}_j - \sum_{i \neq j} \beta^*_{ij} \boldsymbol{x}_i.$$

Then, $\boldsymbol{\varepsilon}_j$ and $\{\boldsymbol{x}_i\}_{i \neq j}$ are independent for every j. Moreover, for every j,

$$\boldsymbol{\varepsilon}_j \sim \mathcal{N}(0, (\sigma^*_j)^2 \boldsymbol{I}_n).$$

4.B.1 Useful Lemmas

Lemma 4.3 (Theorem 1.19, [191])**.** *Let $\omega \in \mathbb{R}^p$ be a random vector with $\boldsymbol{\omega}_i \overset{iid}{\sim} \mathcal{N}(0, \sigma^2)$, then*

$$\mathbb{P}(\sup_{\boldsymbol{\theta} \in \mathcal{B}(p)} \boldsymbol{\theta}^\top \boldsymbol{\omega} > t) \leq \exp\left(-\frac{t^2}{8\sigma^2} + p \log 5\right), \tag{4.58}$$

where $\mathcal{B}(p)$ denotes the unit Euclidean ball of dimension p.

Lemma 4.4 (Lemma 6, [20])**.** *Suppose the rows of the matrix $\boldsymbol{X} \in \mathbb{R}^{n \times p}$ are iid draws from a multivariate Gaussian distribution $\mathcal{N}(0, \boldsymbol{G})$. Moreover, suppose for any $S \subseteq [p]$ such that $|S| \leq k$,*

$$\lambda_{\min}(\boldsymbol{G}_{S,S}) \geq \kappa^2 > 0.$$

Then, if $n \gtrsim \log p$, with probability at least $1 - k \exp(-10k \log p)$, we have:

$$\sigma_{\min}(X_S) \gtrsim \kappa \sqrt{n} \quad \text{for all } S \text{ with } |S| \leq k.$$

(We recall that $\boldsymbol{X}_S$ is a sub-matrix of $\boldsymbol{X}$ restricted to the columns indexed by S).

Lemma 4.5. *Suppose $\boldsymbol{X} \in \mathbb{R}^{n \times p}$ has iid rows of $\mathcal{N}(0, \boldsymbol{G})$. For fixed $j_1, j_2 \in [p]$, we*

have

$$\mathbb{P}\left(\left|\frac{1}{n}\sum_{i=1}^{n} x_{ij_1}x_{ij_2} - g_{j_1j_2}\right| > c_\psi(|g_{j_1j_2}| + \sqrt{g_{j_1j_1}g_{j_2j_2}})\sqrt{\frac{\log(1/\delta)}{C_b n}}\right) \leq 2\delta \tag{4.59}$$

if $n > \frac{2}{C_b}\log(1/\delta)$*, for some absolute constants* $C_b, c_\psi > 0$*.*

Proof. Note that $\mathbb{E}[x_{ij_1}x_{ij_2}] = g_{j_1j_2}$, the ψ_1-Orlicz norm of $x_{ij_1}x_{ij_2} - g_{j_1j_2}$ can be bounded as

$$\|x_{ij_1}x_{ij_2} - g_{j_1j_2}\|_{\psi_1} \leq \|x_{ij_1}\|_{\psi_2}\|x_{ij_2}\|_{\psi_2} + \|g_{j_1j_2}\|_{\psi_1} \leq c_\psi(|g_{j_1j_2}| + \sqrt{g_{j_1j_1}g_{j_2j_2}})$$

for some $c_\psi > 0$. Consequently, by Bernstein's inequality [211, Theorem 2.8.1]

$$\begin{aligned}&\mathbb{P}\left(\left|\frac{1}{n}\sum_{i=1}^{n} x_{ij_1}x_{ij_2} - g_{j_1j_2}\right| > t\right)\\ &\quad\leq 2\exp\left(-C_b n\left[\frac{t^2}{c_\psi^2(|g_{j_1j_2}| + \sqrt{g_{j_1j_1}g_{j_2j_2}})^2} \wedge \frac{t}{c_\psi(|g_{j_1j_2}| + \sqrt{g_{j_1j_1}g_{j_2j_2}})}\right]\right).\end{aligned} \tag{4.60}$$

for some constant $C_b > 0$. Take

$$t = c_\psi(|g_{j_1j_2}| + \sqrt{g_{j_1j_1}g_{j_2j_2}})\sqrt{\frac{\log(1/\delta)}{C_b n}}$$

and

$$n > \frac{2}{C_b}\log(1/\delta).$$

As a result,

$$\frac{t^2}{c_\psi^2(|g_{j_1j_2}| + \sqrt{g_{j_1j_1}g_{j_2j_2}})^2} = \frac{\log(1/\delta)}{C_b n} \leq \sqrt{\frac{\log(1/\delta)}{C_b n}} = \frac{t}{c_\psi(|g_{j_1j_2}| + \sqrt{g_{j_1j_1}g_{j_2j_2}})}$$

so

$$\left[\frac{t^2}{c_\psi^2(|g_{j_1j_2}| + \sqrt{g_{j_1j_1}g_{j_2j_2}})^2} \wedge \frac{t}{c_\psi(|g_{j_1j_2}| + \sqrt{g_{j_1j_1}g_{j_2j_2}})}\right] = \frac{t^2}{c_\psi^2(|g_{j_1j_2}| + \sqrt{g_{j_1j_1}g_{j_2j_2}})^2}$$

which completes the proof with (4.60). □

4.B.2 Proof of Theorem 4.3

Lemma 4.6. *Let*

$$\begin{aligned} \hat{\boldsymbol{y}}_j &= \boldsymbol{x}_j - \sum_{i \neq j} \hat{\beta}_{ij} \boldsymbol{x}_i, \\ \boldsymbol{y}_j^* &= \boldsymbol{x}_j - \sum_{i \neq j} \beta_{ij}^* \boldsymbol{x}_i \end{aligned} \tag{4.61}$$

for $j \in [p]$. Let the event E_1 be defined as

$$E_1 = \left\{ \sum_{j \in [p]} \frac{1}{20} (\hat{\sigma}_j - \sigma_j^*)^2 + \sum_{j \in [p]} \frac{\|\hat{\boldsymbol{y}}_j\|_2^2 - \|\boldsymbol{y}_j^*\|_2^2}{2n\hat{\sigma}_j^2} \lesssim \frac{1}{l_\sigma^2} \frac{p \log(p/k)}{n} \right\}. \tag{4.62}$$

Under the assumptions of the theorem,

$$\mathbb{P}(E_1) \geq 1 - p(k/p)^{10}.$$

Proof. Let the event $\mathcal{E}_j$ be defined as

$$\mathcal{E}_j = \left\{ \left| (\sigma_j^*)^2 - \frac{\|\boldsymbol{y}_j^*\|_2^2}{n} \right| \lesssim (\sigma_j^*)^2 \sqrt{\frac{\log(p/k)}{n}} \right\}.$$

Note that $\|\boldsymbol{y}_j^*\|_2^2 = \|\boldsymbol{\varepsilon}_j\|_2^2$, therefore by taking $\delta = (p/k)^{10}$ in Lemma 4.5, as $n \gtrsim \log p$, one has

$$\left| (\sigma_j^*)^2 - \frac{\|\boldsymbol{y}_j^*\|_2^2}{n} \right| = \left| (\sigma_j^*)^2 - \frac{1}{n} \sum_{i=1}^n (\varepsilon_j)_i^2 \right|$$

so

$$\mathbb{P}\left(\mathcal{E}_j\right) \geq 1 - (k/p)^{10}. \tag{4.63}$$

As a result, by union bound

$$\mathbb{P}\left(\bigcap_{j \in [p]} \mathcal{E}_j \right) \geq 1 - p(k/p)^{10}. \tag{4.64}$$

In particular, note that if we take $n \gtrsim 36\log(p/k)$, we achieve

$$\frac{\|\boldsymbol{y}_j^*\|_2^2}{n} \geq \frac{5(\sigma_j^*)^2}{6}. \tag{4.65}$$

The rest of the proof is on the event $\bigcap_{j\in[p]} \mathcal{E}_j$.

Let

$$f_j(x) = \log(x) + \frac{\|\boldsymbol{y}_j^*\|_2^2}{2n}\frac{1}{x^2}. \tag{4.66}$$

By optimality of $\{\hat{\sigma}_j, \hat{\beta}_{ij}\}$ and feasibility of $\{\sigma_j^*, \beta_{ij}^*\}$ for Problem (4.19),

$$\begin{aligned}
&\sum_{j\in[p]} \log(\hat{\sigma}_j) + \frac{\|\hat{\boldsymbol{y}}_j\|_2^2}{2n\hat{\sigma}_j^2} \leq \sum_{j\in[p]} \log(\sigma_j^*) + \frac{\|\boldsymbol{y}_j^*\|_2^2}{2n(\sigma_j^*)^2} \\
\Rightarrow &\sum_{j\in[p]} \left[\log(\frac{\hat{\sigma}_j}{\sigma_j^*}) + \frac{\|\boldsymbol{y}_j^*\|_2^2}{2n}\left(\frac{1}{\hat{\sigma}_j^2} - \frac{1}{(\sigma_j^*)^2}\right) + \frac{\|\hat{\boldsymbol{y}}_j\|_2^2 - \|\boldsymbol{y}_j^*\|_2^2}{2n\hat{\sigma}_j^2}\right] \leq 0 \\
\Rightarrow &\sum_{j\in[p]} \left[f_j(\hat{\sigma}_j) - f_j(\sigma_j^*)\right] \leq \sum_{j\in[p]} \frac{\|\boldsymbol{y}_j^*\|_2^2 - \|\hat{\boldsymbol{y}}_j\|_2^2}{2n\hat{\sigma}_j^2}.
\end{aligned} \tag{4.67}$$

By (4.66),.

$$\begin{aligned}
f_j'(x) &= \frac{1}{x} - \frac{\|\boldsymbol{y}_j^*\|_2^2}{nx^3}, \\
f_j''(x) &= -\frac{1}{x^2} + \frac{3\|\boldsymbol{y}_j^*\|_2^2}{nx^4}.
\end{aligned} \tag{4.68}$$

Therefore, by Taylor's expansion of f_j,

$$\begin{aligned}
f_j(\hat{\sigma}_j) - f_j(\sigma_j^*) &= \left[\frac{1}{\sigma_j^*} - \frac{\|\boldsymbol{y}_j^*\|_2^2}{n(\sigma_j^*)^3}\right](\hat{\sigma}_j - \sigma_j^*) + \frac{1}{2}\left[-\frac{1}{x^2} + \frac{3\|\boldsymbol{y}_j^*\|_2^2}{nx^4}\right](\hat{\sigma}_j - \sigma_j^*)^2 \\
&\overset{(a)}{\geq} -\left[\frac{1}{\sigma_j^*} - \frac{\|\boldsymbol{y}_j^*\|_2^2}{n(\sigma_j^*)^3}\right]^2 + \frac{1}{2}\left[-\frac{1}{x^2} + \frac{3\|\boldsymbol{y}_j^*\|_2^2}{nx^4} - \frac{1}{2}\right](\hat{\sigma}_j - \sigma_j^*)^2
\end{aligned} \tag{4.69}$$

for some x between σ_j^* and $\hat{\sigma}_j$ where (a) is by the inequality $2ab \geq -2a^2 - b^2/2$.

Consequently, for any $x \in [l_\sigma, u_\sigma]$,

$$\begin{aligned} -\frac{1}{x^2} + \frac{3\|\boldsymbol{y}_j^*\|_2^2}{nx^4} - \frac{1}{2} &= \frac{6\|\boldsymbol{y}_j^*\|_2^2/n - 2x^2 - x^4}{2x^4} \\ &\overset{(a)}{\geq} \frac{5(\sigma_j^*)^2 - 2u_\sigma^2 - u_\sigma^4}{2x^4} \\ &\geq \frac{5l_\sigma^2 - 2u_\sigma^2 - u_\sigma^4}{2x^4} \\ &\geq \frac{5l_\sigma^2 - 2u_\sigma^2 - u_\sigma^4}{2u_\sigma^4} > \frac{1}{10} \end{aligned} \tag{4.70}$$

where the last inequality is due to Assumption 4.1 and (a) is due to (4.65). By substituting (4.69) and (4.70) into (4.67), we obtain

$$\sum_{j\in[p]} \frac{1}{20}(\hat{\sigma}_j - \sigma_j^*)^2 + \sum_{j\in[p]} \frac{\|\hat{\boldsymbol{y}}_j\|_2^2 - \|\boldsymbol{y}_j^*\|_2^2}{2n\hat{\sigma}_j^2} \leq \sum_{j\in[p]} \left[\frac{1}{\sigma_j^*} - \frac{\|\boldsymbol{y}_j^*\|_2^2}{n(\sigma_j^{*3})}\right]^2. \tag{4.71}$$

From (4.63), for any $j \in [p]$

$$\begin{aligned} \left[\frac{1}{\sigma_j^*} - \frac{\|\boldsymbol{y}_j^*\|_2^2}{n(\sigma_j^*)^3}\right]^2 &= \frac{1}{(\sigma_j^*)^6}\left[(\sigma_j^*)^2 - \frac{\|\boldsymbol{y}_j^*\|_2^2}{n}\right]^2 \\ &\lesssim \frac{1}{(\sigma_j^*)^2}\frac{\log(p/k)}{n} \lesssim \frac{1}{l_\sigma^2}\frac{\log(p/k)}{n}. \end{aligned} \tag{4.72}$$

As a result, by substituting (4.72) into (4.71) the proof is complete. □

Lemma 4.7. *Let $\boldsymbol{y}_j^*, \hat{\boldsymbol{y}}_j$ be defined as in* (4.61)*. Let the event E_2 be defined as*

$$E_2 = \left\{ \frac{1}{2nu_\sigma^2} \sum_{j\in[p]} \left\| \sum_{i:i\neq j} (\beta_{ij}^* - \hat{\beta}_{ij})\boldsymbol{x}_i \right\|_2^2 \lesssim \sum_{j\in[p]} \frac{\|\hat{\boldsymbol{y}}_j\|_2^2 - \|\boldsymbol{y}_j^*\|_2^2}{n\hat{\sigma}_j^2} + \frac{u_\sigma^2 kp\log(2p/k)}{l_\sigma^2 n} \right\}.$$

Under the assumptions of Theorem 4.3,

$$\mathbb{P}(E_2) \geq 1 - p\exp(-10k\log(p/k)).$$

Proof. One has

$$
\begin{aligned}
&\|\hat{\boldsymbol{y}}_j\|_2^2 - \|\boldsymbol{y}_j^*\|_2^2 \\
&= \left\| \sum_{i:i\neq j} (\beta_{ij}^* - \hat{\beta}_{ij}) \boldsymbol{x}_i + \boldsymbol{\varepsilon}_j \right\|_2^2 - \|\boldsymbol{\varepsilon}_j\|_2^2 \\
&= \left\| \sum_{i:i\neq j} (\beta_{ij}^* - \hat{\beta}_{ij}) \boldsymbol{x}_i \right\|_2^2 + 2\boldsymbol{\varepsilon}_j^\top \sum_{i\neq j} (\beta_{ij}^* - \hat{\beta}_{ij}) \boldsymbol{x}_i \\
&= \left\| \sum_{i:i\neq j} (\beta_{ij}^* - \hat{\beta}_{ij}) \boldsymbol{x}_i \right\|_2^2 + 2\boldsymbol{\varepsilon}_j^\top \frac{\sum_{i\neq j} (\beta_{ij}^* - \hat{\beta}_{ij}) \boldsymbol{x}_i}{\| \sum_{i:i\neq j} (\beta_{ij}^* - \hat{\beta}_{ij}) \boldsymbol{x}_i \|_2} \| \sum_{i\neq j} (\beta_{ij}^* - \hat{\beta}_{ij}) \boldsymbol{x}_i \|_2 \\
&\overset{(a)}{\geq} \frac{1}{2} \left\| \sum_{i:i\neq j} (\beta_{ij}^* - \hat{\beta}_{ij}) \boldsymbol{x}_i \right\|_2^2 - 2 \left(\boldsymbol{\varepsilon}_j^\top \frac{\sum_{i:i\neq j} (\beta_{ij}^* - \hat{\beta}_{ij}) \boldsymbol{x}_i}{\| \sum_{i:i\neq j} (\beta_{ij}^* - \hat{\beta}_{ij}) \boldsymbol{x}_i \|_2} \right)^2
\end{aligned} \tag{4.73}
$$

where (a) is by the inequality $2ab \geq -2a^2 - b^2/2$. For $j \in [p]$ let

$$
\begin{aligned}
\hat{S}_j &= \{i \in [p] : i \neq j, \hat{\beta}_{ij} \neq 0\}, \\
S_j^* &= \{i \in [p] : i \neq j, \beta_{ij}^* \neq 0\}.
\end{aligned} \tag{4.74}
$$

Moreover, let $S_j = \hat{S}_j \cup S_j^*$. Note that $|S_j| \leq 2k$. Suppose $\boldsymbol{\Phi}_S \in \mathbb{R}^{n\times|S|}$ is an orthonormal basis for the column span of $\boldsymbol{X}_S$ for $S \subseteq [p]$. By Lemma 4.2, if $j \notin S$, then $\boldsymbol{\varepsilon}_j$ and $\boldsymbol{X}_{[p]\setminus\{j\}}$ are independent. As a result, we have the conditional distribution

$\boldsymbol{\Phi}_S^\top \boldsymbol{\varepsilon}_j | \boldsymbol{X}_{[p]\setminus\{j\}} \sim \mathcal{N}(0, (\sigma_j^*)^2 \boldsymbol{I}_{|S|})$. Given this fact, one has for $t > 0$ and a fixed $j \in [p]$,

$$
\begin{aligned}
&\mathbb{P}\left(\left(\boldsymbol{\varepsilon}_j^\top \frac{\sum_{i:i\neq j}(\beta_{ij}^* - \hat{\beta}_{ij})\boldsymbol{x}_i}{\|\sum_{i:i\neq j}(\beta_{ij}^* - \hat{\beta}_{ij})\boldsymbol{x}_i\|_2}\right)^2 > t\right) \\
&\leq \mathbb{P}\left(\sup_{\substack{\boldsymbol{v}\in\mathbb{R}^p \\ S(\boldsymbol{v})=S_j}} \left(\boldsymbol{\varepsilon}_j^\top \frac{\boldsymbol{X}\boldsymbol{v}}{\|\boldsymbol{X}\boldsymbol{v}\|_2}\right)^2 > t\right) \\
&\leq \mathbb{P}\left(\max_{\substack{S\subseteq[p]\setminus\{j\} \\ |S|=2k}} \sup_{\boldsymbol{v}\in\mathbb{R}^{2k}} \left(\boldsymbol{\varepsilon}_j^\top \frac{\boldsymbol{X}_S\boldsymbol{v}}{\|\boldsymbol{X}_S\boldsymbol{v}\|_2}\right)^2 > t\right) \\
&\overset{(a)}{=} \mathbb{P}\left(\max_{\substack{S\subseteq[p]\setminus\{j\} \\ |S|=2k}} \sup_{\boldsymbol{v}\in\mathbb{R}^{2k}} \left(\boldsymbol{\varepsilon}_j^\top \frac{\boldsymbol{X}_S\boldsymbol{v}}{\|\boldsymbol{X}_S\boldsymbol{v}\|_2}\right)^2 > t \,\middle|\, \boldsymbol{X}_{[p]\setminus\{j\}}\right) \\
&\overset{(b)}{\leq} \mathbb{P}\left(\max_{\substack{S\subseteq[p]\setminus\{j\} \\ |S|=2k}} \sup_{\boldsymbol{\alpha}\in\mathcal{B}(2k)} \left(\boldsymbol{\varepsilon}_j^\top \boldsymbol{\Phi}_S\boldsymbol{\alpha}\right)^2 > t \,\middle|\, \boldsymbol{X}_{[p]\setminus\{j\}}\right) \\
&\leq \sum_{\substack{S\subseteq[p]\setminus\{j\} \\ |S|=2k}} \mathbb{P}\left(\sup_{\boldsymbol{\alpha}\in\mathcal{B}(2k)} \left(\boldsymbol{\varepsilon}_j^\top \boldsymbol{\Phi}_S\boldsymbol{\alpha}\right)^2 > t \,\middle|\, \boldsymbol{X}_{[p]\setminus\{j\}}\right) \\
&\overset{(c)}{\leq} \sum_{\substack{S\subseteq[p]\setminus\{j\} \\ |S|=2k}} \exp\left(-\frac{t}{8(\sigma_j^*)^2} + 2k\log 5\right) \\
&\overset{(d)}{\leq} \left(\frac{ep}{2k}\right)^{2k} \exp\left(-\frac{t}{8(\sigma_j^*)^2} + 2k\log 5\right) \\
&\leq \exp\left(-\frac{t}{8u_\sigma^2} + 4k\log(2p/k)\right)
\end{aligned} \tag{4.75}
$$

where (a) is due to independence of $\boldsymbol{\varepsilon}_j$ and $\boldsymbol{X}_{[p]\setminus\{j\}}$ as discussed above, (b) is true as $\boldsymbol{X}_S\boldsymbol{v}$ is in the column span of $\boldsymbol{\Phi}_S$, (c) is due to Lemma 4.3 and the conditional distribution discussed above and (d) is due to the inequality $\binom{n}{k} \leq (ep/k)^k$. Take

$$
t = 8cu_\sigma^2 s\log(2p/k)
$$

so from (4.75),

$$\mathbb{P}\left(\left(\boldsymbol{\varepsilon}_j^\top \frac{\sum_{i:i\neq j}(\beta^*_{ij}-\hat{\beta}_{ij})\boldsymbol{x}_i}{\|\sum_{i:i\neq j}(\beta^*_{ij}-\hat{\beta}_{ij})\boldsymbol{x}_i\|_2}\right)^2 > 8cu_\sigma^2 k\log(2p/k)\right) \leq \exp\left(-(c-4)k\log(2p/k)\right). \tag{4.76}$$

Take c sufficiently large and by union bound over $j\in[p]$, we have

$$\mathbb{P}\left(\sum_{j=1}^p\left(\boldsymbol{\varepsilon}_j^\top \frac{\sum_{i:i\neq j}(\beta^*_{ij}-\hat{\beta}_{ij})\boldsymbol{x}_i}{\|\sum_{i:i\neq j}(\beta^*_{ij}-\hat{\beta}_{ij})\boldsymbol{x}_i\|_2}\right)^2 \lesssim u_\sigma^2 kp\log(2p/k)\right) \geq 1-p\exp(-10k\log(p/k)). \tag{4.77}$$

By (4.73) and (4.77),

$$\begin{aligned}
&\sum_{j\in[p]}\frac{1}{2nu_\sigma^2}\left\|\sum_{i:i\neq j}(\beta^*_{ij}-\hat{\beta}_{ij})\boldsymbol{x}_i\right\|_2^2 \\
&\leq \sum_{j\in[p]}\frac{1}{2n\hat{\sigma}_j^2}\left\|\sum_{i:i\neq j}(\beta^*_{ij}-\hat{\beta}_{ij})\boldsymbol{x}_i\right\|_2^2 \\
&\leq \sum_{j\in[p]}\left[\frac{\|\hat{\boldsymbol{y}}_j\|_2^2-\|\boldsymbol{y}_j^*\|_2^2}{n\hat{\sigma}_j^2}+\frac{2}{n\hat{\sigma}_j^2}\left(\boldsymbol{\varepsilon}_j^\top\frac{\sum_{i:i\neq j}(\beta^*_{ij}-\hat{\beta}_{ij})\boldsymbol{x}_i}{\|\sum_{i:i\neq j}(\beta^*_{ij}-\hat{\beta}_{ij})\boldsymbol{x}_i\|_2}\right)^2\right] \\
&\lesssim \sum_{j\in[p]}\frac{\|\hat{\boldsymbol{y}}_j\|_2^2-\|\boldsymbol{y}_j^*\|_2^2}{n\hat{\sigma}_j^2}+\frac{u_\sigma^2 kp\log(2p/k)}{l_\sigma^2 n}
\end{aligned}$$

with high probability. □

Proof of Theorem 4.3. Let us define the events A_j for $j\in[p]$ as

$$A_j = \left\{\sigma_{\min}(\boldsymbol{X}_S)\gtrsim\sqrt{n} : S\subseteq[p]\setminus\{j\}, |S|\leq 2k\right\}.$$

By Lemma 4.4 and part 5 of Assumption 4.1, we have $\mathbb{P}(A_j)\geq 1-k\exp(-10k\log(p-1))$ so by union bound over $j\in[p]$,

$$\mathbb{P}(\cap_{j\in[p]}A_j)\geq 1-2kp\exp(-20k\log(p-1)).$$

The rest of the proof is on the intersection of events E_1, E_2 from Lemmas 4.6 and 4.7

and $\cap_j A_j$. By Lemmas 4.6 and 4.7, this happens with probability at least

$$1 - 2kp\exp(-20k\log(p-1)) - p(k/p)^{10} - p\exp(-10k\log(p/k)). \tag{4.78}$$

One has

$$\begin{aligned}
\frac{1}{n}\sum_{\in[p]}\left\|\sum_{i:i\neq j}(\beta^*_{ij}-\hat{\beta}_{ij})\boldsymbol{x}_i\right\|_2^2 &= \frac{1}{n}\sum_{j\in[p]}\left\|\boldsymbol{X}_{S_j}(\hat{\boldsymbol{\beta}}_{S_j,j}-\boldsymbol{\beta}^*_{S_j,j})\right\|_2^2 \\
&\geq \frac{1}{n}\sum_{j\in[p]}\sigma^2_{\min}(\boldsymbol{X}_{S_j})\left\|\hat{\boldsymbol{\beta}}_{S_j,j}-\boldsymbol{\beta}^*_{S_j,j}\right\|_2^2 \\
&\gtrsim \sum_{j\in[p]}\sum_{i:i\neq j}(\beta^*_{ij}-\hat{\beta}_{ij})^2
\end{aligned} \tag{4.79}$$

where $\boldsymbol{\beta}_{S_j,j} \in \mathbb{R}^{|S_j|}$ is the vector containing the values $\{\beta_{i,j}$ for $i \in S_j$. The last inequality above is a result of event A_j. As a result,

$$\begin{aligned}
&\sum_{j\in[p]}(\hat{\sigma}_j-\sigma^*_j)^2 + \frac{1}{u_\sigma^2}\sum_{j\in[p]}\sum_{i:i\neq j}(\beta^*_{ij}-\hat{\beta}_{ij})^2 \\
&\overset{(a)}{\lesssim} \sum_{j\in[p]}(\hat{\sigma}_j-\sigma^*_j)^2 + \frac{1}{nu_\sigma^2}\sum_{\in[p]}\left\|\sum_{i:i\neq j}(\beta^*_{ij}-\hat{\beta}_{ij})\boldsymbol{x}_i\right\|_2^2 \\
&\overset{(b)}{\lesssim} \sum_{j\in[p]}(\hat{\sigma}_j-\sigma^*_j)^2 + \sum_{j\in[p]}\frac{\|\hat{\boldsymbol{y}}_j\|_2^2-\|\boldsymbol{y}^*_j\|_2^2}{n\hat{\sigma}_j^2} + \frac{u_\sigma^2 kp\log(2p/k)}{l_\sigma^2 n} \\
&\overset{(c)}{\lesssim} \frac{u_\sigma^2 kp\log(2p/k)}{l_\sigma^2 n}
\end{aligned}$$

where (a) is due to (4.79), (b) is due to Lemma 4.7 and (c) is due to Lemma 4.6. $\square$

4.B.3 Proof of Theorem 4.4

Proof. Based on the definition of $\mathbf{\Theta}^*, \hat{\mathbf{\Theta}}$ for $i \neq j \in [p]$ one has

$$\begin{aligned}
|\hat{\theta}_{ji} - \theta^*_{ji}| &= \left| \frac{\hat{\beta}_{ij}}{\hat{\sigma}_j^2} - \frac{\beta^*_{ij}}{(\sigma^*_j)^2} \right| \\
&\leq \frac{|\hat{\beta}_{ij}(\sigma^*_j)^2 - \beta^*_{ij}\hat{\sigma}_j^2|}{l_\sigma^4} \\
&\leq \frac{|(\hat{\beta}_{ij} - \beta^*_{ij})(\sigma^*_j)^2| + |\beta^*_{ij}(\hat{\sigma}_j^2 - (\sigma^*_j)^2)|}{l_\sigma^4} \\
&\leq \frac{|(\hat{\beta}_{ij} - \beta^*_{ij})(\sigma^*_j)^2|}{l_\sigma^4} + \frac{|\beta^*_{ij}||\hat{\sigma}_j - \sigma^*_j||\hat{\sigma}_j + \sigma^*_j|}{l_\sigma^4} \\
&\lesssim \frac{|\hat{\beta}_{ij} - \beta^*_{ij}|u_\sigma^2}{l_\sigma^4} + \frac{|\hat{\sigma}_j - \sigma^*_j|u_\sigma^2}{l_\sigma^4}
\end{aligned} \tag{4.80}$$

where the last inequality is due to Assumption 4.1. Similarly,

$$\begin{aligned}
|\hat{\theta}_{jj} - \theta^*_{jj}| &= \left| \frac{1}{\hat{\sigma}_j^2} - \frac{1}{(\sigma^*_j)^2} \right| \\
&\leq \frac{|(\sigma^*_j)^2 - \hat{\sigma}_j^2|}{l_\sigma^4} \\
&\lesssim \frac{|\hat{\sigma}_j - \sigma^*_j|u_\sigma^2}{l_\sigma^4}.
\end{aligned} \tag{4.81}$$

As a result,

$$\begin{aligned}
\left\| \hat{\mathbf{\Theta}} - \mathbf{\Theta}^* \right\|_F^2 &= \sum_{i \neq j \in [p]} |\hat{\theta}_{ji} - \theta^*_{ji}|^2 + \sum_{j \in [p]} |\hat{\theta}_{jj} - \theta^*_{jj}|^2 \\
&\overset{(a)}{\lesssim} \frac{u_\sigma^4}{l_\sigma^8} \left[\sum_{i \neq j \in [p]} |\hat{\beta}_{ij} - \beta^*_{ij}|^2 + \sum_{j \in [p]} |\hat{\sigma}_j - \sigma^*_j|^2 \right] \\
&\overset{(b)}{\lesssim} \frac{(u_\sigma^6 + u_\sigma^8) kp \log(2p/k)}{l_\sigma^{10} n}
\end{aligned} \tag{4.82}$$

with high probability, where (a) is due to (4.80) and (4.81) and (b) is because of Theorem 4.3. □

4.B.4 Proof of Theorem 4.5

We first introduce some notation that we will be using in this proof.

Notation. For $S \subseteq [p]$, we denote the projection matrix onto the column span of $\boldsymbol{X}_S$ by $\boldsymbol{P}_{\boldsymbol{X}_S}$. Note that if $\boldsymbol{X}_S$ has linearly independent columns, $\boldsymbol{P}_{\boldsymbol{X}_S} = \boldsymbol{X}_S(\boldsymbol{X}_S^\top \boldsymbol{X}_S)^{-1}\boldsymbol{X}_S^\top$. In our case, as the data is drawn from a normal distribution with a full-rank covariance matrix, for any $S \subseteq [p]$ with $|S| < n$, $\boldsymbol{X}_S$ has linearly independent columns with probability one. We define the operator of $\boldsymbol{A} \in \mathbb{R}^{p_1 \times p_2}$ as

$$\|\boldsymbol{A}\|_{\text{op}} = \max_{\substack{\boldsymbol{x} \in \mathbb{R}^{p_1} \\ \boldsymbol{x} \neq 0}} \frac{\|\boldsymbol{A}\boldsymbol{x}\|_2}{\|\boldsymbol{x}\|_2}.$$

The solution to the least squares problem with the support restricted to S,

$$\min_{\boldsymbol{\beta}_{S^c}=0} \frac{1}{n}\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|_2^2 \tag{4.83}$$

for $\boldsymbol{y} \in \mathbb{R}^n$ and $\boldsymbol{X} \in \mathbb{R}^{n \times p}$ is given by

$$\boldsymbol{\beta}_S = (\boldsymbol{X}_S^\top \boldsymbol{X}_S)^{-1}\boldsymbol{X}_S^\top \boldsymbol{y}.$$

Note that as in our case the data is drawn from normal distribution with a full-rank covariance matrix, $(\boldsymbol{\beta}_S)_i \neq 0$ for $i \in S$. Consequently, we denote the optimal objective in (4.83) by

$$\mathcal{L}_S(\boldsymbol{y}) = \frac{1}{n}\boldsymbol{y}^\top(\boldsymbol{I}_n - \boldsymbol{P}_{\boldsymbol{X}_S})\boldsymbol{y}. \tag{4.84}$$

For $S_1, S_2 \subseteq [p]$, $\boldsymbol{\Sigma} \in \mathbb{R}^{p \times p}$ positive definite and $S_0 = S_2 \setminus S_1$, we let

$$\boldsymbol{\Sigma}/[S_1, S_2] = \boldsymbol{\Sigma}_{S_0,S_0} - \boldsymbol{\Sigma}_{S_0,S_1}\boldsymbol{\Sigma}_{S_1,S_1}^{-1}\boldsymbol{\Sigma}_{S_1,S_0}. \tag{4.85}$$

Note that $\boldsymbol{\Sigma}/[S_1, S_2]$ is the Schur complement of the matrix

$$\boldsymbol{\Sigma}(S_1, S_2) = \begin{bmatrix} \boldsymbol{\Sigma}_{S_1,S_1} & \boldsymbol{\Sigma}_{S_1,S_0} \\ \boldsymbol{\Sigma}_{S_0,S_1} & \boldsymbol{\Sigma}_{S_0,S_0} \end{bmatrix}. \tag{4.86}$$

Let $S_j^*, \hat{S}_j, t_j, \tilde{t}_j$ for $j \in [p]$ be defined as

$$\begin{aligned}
\hat{S}_j &= \{i \in [p] : i \neq j, \hat{\beta}_{ij} \neq 0\}, \\
S_j^* &= \{i \in [p] : i \neq j, \beta_{ij}^* \neq 0\}, \\
t_j &= |S_j^* \setminus \hat{S}_j|, \\
\bar{t}_j &= |\hat{S}_j \setminus S_j^*|, \\
\tilde{t}_j &= \left|(S_j^* \setminus \hat{S}_j) \cap \{j+1, \cdots, p\}\right|.
\end{aligned} \tag{4.87}$$

Let us define for $j \in [p]$,

$$h_j(\sigma, S) = \log(\sigma) + \frac{\mathcal{L}_S(\boldsymbol{x}_j)}{2\sigma^2}. \tag{4.88}$$

Roadmap: At optimality of Problem (4.27), the optimal objective is given as

$$\sum_{j=1}^{p} \left\{ h_j(\hat{\sigma}_j, \hat{S}_j) + \lambda |\hat{S}_j| \right\}.$$

Similarly, if we fix the value of z_{ij} to z_{ij}^*, the objective value is

$$\sum_{j=1}^{p} \left\{ h_j(\tilde{\sigma}_j, S_j^*) + \lambda |S_j^*| \right\}$$

where $\tilde{\sigma}_j$ are optimal variance values from Problem (4.27) on the underlying support. Next, we divide variables into two parts based on the value of $\mathcal{L}_{\hat{S}_j}$:

$$\mathcal{J} = \left\{ j \in [p] : \mathcal{L}_{\hat{S}_j}(\boldsymbol{x}_j) \geq \ell \right\} \tag{4.89}$$

We note that the function

$$f(x) = \log(x) + \frac{a}{2x^2}$$

for $a, x > 0$ is minimized for $x^2 = a$. Therefore, for $j \in \mathcal{J}$ we have $\hat{\sigma}_j^2 = \mathcal{L}_{\hat{S}_j}(\boldsymbol{x}_j) \geq \ell$. This leads to $h_j(\hat{\sigma}_j, \hat{S}_j) = \log(\mathcal{L}_{\hat{S}_j}(\boldsymbol{x}_j))/2 + 1/2$. Moreover,

$$f'(x) = \frac{1}{x} - \frac{a}{x^3} = \frac{1}{x^3}(x^2 - a) \geq 0$$

for $x \geq \sqrt{a}$, showing $f(x)$ is minimized for $x = \sqrt{a}$ for $x \geq \sqrt{a}$. As a result, for $j \in \mathcal{J}^c$, $\hat{\sigma}_j^2 = \ell$. As a result, the optimal objective of Problem (4.27) is given as

$$\begin{aligned}
&\sum_{j=1}^{p} \left\{ h_j(\hat{\sigma}_j, \hat{S}_j) + \lambda |\hat{S}_j| \right\} \\
&= \sum_{j \in \mathcal{J}^c} \left\{ h_j(\sqrt{\ell}, \hat{S}_j) + \lambda |\hat{S}_j| \right\} + \sum_{j \in \mathcal{J}} \left\{ h_j(\sqrt{\mathcal{L}_{\hat{S}_j}(\boldsymbol{x}_j)}, \hat{S}_j) + \lambda |\hat{S}_j| \right\} \\
&= \sum_{j \in \mathcal{J}^c} \left\{ h_j(\sqrt{\ell}, \hat{S}_j) + \lambda |\hat{S}_j| \right\} + \sum_{j \in \mathcal{J}} \left\{ \frac{1}{2} \log(\mathcal{L}_{\hat{S}_j}(\boldsymbol{x}_j)) + \frac{1}{2} + \lambda |\hat{S}_j| \right\}. \qquad (4.90)
\end{aligned}$$

We also will show the optimal cost on the correct support is

$$\sum_{j=1}^{p} \left\{ h_j(\tilde{\sigma}_j, S_j^*) + \lambda |S_j^*| \right\} = \sum_{j=1}^{p} \left\{ \frac{1}{2} \log(\mathcal{L}_{S_j^*}(\boldsymbol{x}_j)) + \frac{1}{2} + \lambda |S_j^*| \right\} \qquad (4.91)$$

Our roadmap for this proof is to show that first, $\mathcal{J}^c = \emptyset$ and second, for $j \in \mathcal{J}$, the support is estimated correctly by comparing the objective value of optimal and correct support.

We note that by Assumption 4.2 part **(B5)**, $t_j \leq k$. Let us define the following basic

events for $j \in [p]$ and $S \subseteq [p]$:

$$
\begin{aligned}
\mathcal{E}_1(j,S) &= \left\{ (\boldsymbol{\beta}^*_{S_j^0,j})^\top (\hat{\boldsymbol{\Sigma}}/[S,S_j^*]) \boldsymbol{\beta}^*_{S_j^0,j} \geq 0.2\eta \frac{|\tilde{S}_j^0| \log p}{n} \right\} \\
\mathcal{E}_2(j,S) &= \left\{ \frac{1}{n} \left\| \boldsymbol{X}_{S_j^0} \boldsymbol{\beta}^*_{S_j^0,j} \right\|_2^2 \leq 4 \frac{|S_j^0|}{k} \right\} \\
\mathcal{E}_3(j,S) &= \left\{ \frac{1}{n} \boldsymbol{\varepsilon}_j^\top (\boldsymbol{I}_n - \boldsymbol{P}_{\boldsymbol{X}_S}) \boldsymbol{X}_{S_j^0} \boldsymbol{\beta}^*_{S_j^0,j} \geq \right. \\
&\qquad \left. - c_{t_1} \sigma_j^* \sqrt{(\boldsymbol{\beta}^*_{S_j^0,j})^\top (\hat{\boldsymbol{\Sigma}}/[S,S_j^*]) \boldsymbol{\beta}^*_{S_j^0,j}} \sqrt{\frac{(|S_j^* \setminus S| + |S \setminus S_j^*|) \log p}{n}} \right\} \\
\mathcal{E}_4(j,S) &= \left\{ \boldsymbol{\varepsilon}_j^\top (\boldsymbol{P}_{\boldsymbol{X}_S} - \boldsymbol{P}_{\boldsymbol{X}_{S_j^*}}) \boldsymbol{\varepsilon}_j \leq c_{t_2} (\sigma_j^*)^2 (|S_j^* \setminus S| + |S \setminus S_j^*|) \log p \right\} \\
\mathcal{E}_5(j,S) &= \left\{ -c_{t_3} (\sigma_j^*)^2 k \log p \leq \boldsymbol{\varepsilon}_j^\top \boldsymbol{P}_{\boldsymbol{X}_S} \boldsymbol{\varepsilon}_j \leq c_{t_3} (\sigma_j^*)^2 k \log p \right\} \\
\mathcal{E}_6(j) &= \left\{ \mathcal{L}_{\hat{S}_j}(\boldsymbol{x}_j) \geq \mathcal{L}_{S_j^*}(\boldsymbol{x}_j) + \frac{3}{20} \eta \tilde{t}_j \frac{\log p}{n} - (\sigma_j^*)^2 \frac{(t_j + \bar{t}_j) \log p}{n} (4c_{t_1}^2 + c_{t_2}) - \frac{4t_j}{k} \right\} \\
\mathcal{E}_7(j) &= \left\{ \ell < \frac{2}{3} (\sigma_j^*)^2 \leq \mathcal{L}_{S_j^*}(\boldsymbol{x}_j) \leq \frac{4}{3} (\sigma_j^*)^2 \right\}
\end{aligned}
\tag{4.92}
$$

for some numerical constants $c_{t_1}, c_{t_2}, c_{t_3} > 0$, where $S_j^0 = S_j^* \setminus S$, $\tilde{S}_j^0 = S_j^0 \cap \{j+1, \cdots, p\}$, $\boldsymbol{\beta}^*_{S_j^0,j}$ is the vector $\{\beta^*_{ij}\}_{i \in S_j^0}$ and $\hat{\boldsymbol{\Sigma}} = \boldsymbol{X}^\top \boldsymbol{X}/n$. The following lemmas establish that the events defined above hold with high probability. The proof of some of these results are similar to results shown in [20], building and improving upon the results of [82].

Lemma 4.8. *Under the assumptions of Theorem 4.5, we have*

$$
\mathbb{P}\left(\bigcap_{j \in [p]} \bigcap_{\substack{S_j \subseteq [p] \setminus \{j\} \\ |S_j| \leq k}} \mathcal{E}_1(j, S_j) \right) \geq 1 - p^{-8}. \tag{4.93}
$$

and

$$
\mathbb{P}\left(\bigcap_{j \in [p]} \bigcap_{S_j \subseteq [p] \setminus \{j\}} \mathcal{E}_2(j, S_j) \right) \geq 1 - p^{-8}. \tag{4.94}
$$

Proof. Let the events $\mathcal{E}_0(S)$ for $S \subseteq [p]$ with $|S| \leq 2k$ and $\mathcal{E}_0$ be defined as

$$\mathcal{E}_0(S) = \left\{ \left\| \hat{\mathbf{\Sigma}}_{S,S} - \mathbf{\Sigma}^*_{S,S} \right\|_{\text{op}} \lesssim \sqrt{\frac{k \log p}{n}} \right\},$$
$$\mathcal{E}_0 = \bigcap_{\substack{S \subseteq [p] \\ |S| \leq 2k}} \mathcal{E}_0(S).$$

One has (for example, by Theorem 5.7 of [191] with $\delta = \exp(-11k \log p)$)

$$\mathbb{P}(\mathcal{E}_0(S)) \geq 1 - \exp(-11k \log p)$$

as $n = c_n k \log p$ is sufficiently large and by Assumption 4.2 part **(B6)**, $\|\mathbf{\Sigma}^*_{S,S}\|_{\text{op}} \lesssim 1$. As a result, by union bound

$$\begin{aligned}\mathbb{P}(\mathcal{E}_0) &\geq 1 - \sum_{\substack{S \subseteq [p] \\ |S| \leq 2k}} (1 - \mathbb{P}(\mathcal{E}_0(S))) \geq 1 - \sum_{t=1}^{2k} \binom{p}{t} \exp(-11k \log p) \\ &\geq 1 - \sum_{t=1}^{2k} p^{2k} p^{-11k} \geq 1 - p \times p^{-9} = 1 - p^{-8}.\end{aligned}$$

The rest of the proof is on event $\mathcal{E}_0$. We first consider the proof of (4.93). Consequently, as $|S_j|, |S_j^*| \leq k$,

$$\|\hat{\mathbf{\Sigma}}(S, S_j^*) - \mathbf{\Sigma}^*(S, S_j^*)\|_{\text{op}} \leq c_b \sqrt{\frac{k \log p}{n}} := \pi \tag{4.95}$$

for some constant $c_b > 0$ where $\mathbf{\Sigma}(S_1, S_2)$ is defined in (4.86). Let c_n be sufficiently large such that $\pi < 0.1$. Therefore, one has

$$\begin{aligned}\lambda_{\min}(\hat{\mathbf{\Sigma}}/[S, S_j^*]) &\overset{(a)}{\geq} \lambda_{\min}(\hat{\mathbf{\Sigma}}(S, S_j^*)) \\ &\overset{(b)}{\geq} \lambda_{\min}(\mathbf{\Sigma}^*(S, S_j^*)) - \|\hat{\mathbf{\Sigma}}(S, S_j^*) - \mathbf{\Sigma}^*(S, S_j^*)\|_{\text{op}} \\ &\geq \kappa^2 - 0.1 > 0.2\end{aligned}$$

where (a) is by Corollary 2.3 of [228], (b) is due to Weyl's inequality and the last

inequality is by part **(B6)** of Assumption 4.2. Finally,

$$(\boldsymbol{\beta}^*_{S^0_j,j})^\top(\hat{\boldsymbol{\Sigma}}/[S,S^*_j])\boldsymbol{\beta}^*_{S^0_j,j} \geq \lambda_{\min}(\hat{\boldsymbol{\Sigma}}/[S,S^*_j])\|\boldsymbol{\beta}^*_{S^0_j,j}\|_2^2 \geq 0.2\|\boldsymbol{\beta}^*_{S^0_j,j}\|_2^2 \geq 0.2\eta\frac{|\tilde{S}^0_j|\log p}{n}$$

where the last inequality is achieved by substituting $\beta_{\min}$ condition from Assumption 4.2 part **(B4)**. This completes the proof of (4.93).
We now proceed to prove (4.94). Note that by Weyl's inequality,

$$\lambda_{\max}(\hat{\boldsymbol{\Sigma}}_{S^0_j,S^0_j}) \leq \lambda_{\max}(\boldsymbol{\Sigma}^*_{S^0_j,S^0_j}) + \|\boldsymbol{\Sigma}^*_{S^0_j,S^0_j} - \hat{\boldsymbol{\Sigma}}_{S^0_j,S^0_j}\|_{\rm op} \leq \lambda_{\max}(\boldsymbol{\Sigma}^*_{S^0_j,S^0_j}) + 0.1 \leq 4$$

where the second inequality is due to event $\mathcal{E}_0$ (note that $|S^0_j| \leq k$) and the last inequality is due to part **(B6)** of Assumption 4.2. Finally, note that

$$\frac{1}{n}\left\|\boldsymbol{X}_{S^0_j}\boldsymbol{\beta}^*_{S^0_j,j}\right\|_2^2 = (\boldsymbol{\beta}^*_{S^0_j,j})^\top\frac{\boldsymbol{X}^\top_{S^0_j}\boldsymbol{X}_{S^0_j}}{n}\boldsymbol{\beta}^*_{S^0_j,j} \leq \lambda_{\max}(\hat{\boldsymbol{\Sigma}}_{S^0_j,S^0_j})\|\boldsymbol{\beta}^*_{S^0_j,j}\|_2^2 \leq 4\|\boldsymbol{\beta}^*_{S^0_j,j}\|_2^2 \leq 4\frac{|S^0_j|}{k}$$

where the last inequality is due to part **(B2)** of Assumption 4.2. □

Lemma 4.9. *One has*

$$\mathbb{P}\left(\bigcap_{j\in[p]}\bigcap_{S\subseteq[p]\setminus\{j\}}\mathcal{E}_3(j,S)\right) \geq 1 - 2kp^{-7}. \tag{4.96}$$

Proof. The proof follows a similar path to the proof of Lemma 13 of [20]. Fix $j \in [p], S \subseteq [p]\setminus\{j\}$, and let $t = |S^*_j \setminus S|, \bar{t} = |S \setminus S^*_j|, S^0_j = S^*_j \setminus S$. Note that if $t = 0$, the lemma is trivial. Therefore, without loss of generality we assume $t \geq 1$. Let

$$\boldsymbol{\gamma}^{(j,S)} = (\boldsymbol{I}_n - \boldsymbol{P}_{\boldsymbol{X}_S})\boldsymbol{X}_{S^0_j}\boldsymbol{\beta}^*_{S^0_j,j}.$$

Following the same calculations in Lemma 13 of [20], one has

$$\mathbb{P}\left(\frac{\boldsymbol{\varepsilon}_j^\top\boldsymbol{\gamma}^{(j,S)}}{\|\boldsymbol{\gamma}^{(j,S)}\|_2} < -x\right) \leq \exp\left(-\frac{x^2}{8(\sigma^*_j)^2} + t\log 5\right). \tag{4.97}$$

Take

$$x^2 = 8\xi^2(\sigma_j^*)^2(t+\bar{t})\log p$$

for some absolute constant $\xi > 0$ that is sufficiently large, and noting

$$\|\boldsymbol{\gamma}^{(j,S)}\|_2 = \sqrt{(\boldsymbol{\beta}^*_{S_j^0,j})^\top(\hat{\boldsymbol{\Sigma}}/[S,S_j^*])\boldsymbol{\beta}^*_{S_j^0,j}},$$

we achieve

$$\mathbb{P}\left(\frac{\boldsymbol{\varepsilon}_j^\top\boldsymbol{\gamma}^{(j,S)}}{n} < -\sqrt{8}\xi\sigma_j^*\sqrt{(\boldsymbol{\beta}^*_{S_j^0,j})^\top(\hat{\boldsymbol{\Sigma}}/[S,S_j^*])\boldsymbol{\beta}^*_{S_j^0,j}}\sqrt{\frac{(t+\bar{t})\log p}{n}}\right) \leq \exp(-10(t+\bar{t})\log p). \tag{4.98}$$

Finally, we complete the proof by using union bound over all possible choices of j, t, S. As a result, the probability of the desired event in the lemma being violated is bounded as

$$\begin{aligned}
&\sum_{j=1}^{p}\sum_{t=1}^{k}\sum_{\bar{t}=0}^{p-k}\sum_{\substack{S\subseteq[p]\setminus\{j\}\\|S_j^*\setminus S|=t\\|S\setminus S_j^*|=\bar{t}}}\exp(-10(t+\bar{t})\log p)\\
&=\sum_{j=1}^{p}\sum_{t=1}^{k}\sum_{\bar{t}=0}^{p-k}\binom{k}{t}\binom{p-k}{\bar{t}}\exp(-10(t+\bar{t})\log p)\\
&\leq p\sum_{t=1}^{k}\sum_{\bar{t}=0}^{p}p^t p^{\bar{t}}\exp(-10(t+\bar{t})\log p)\\
&\leq p\sum_{t=1}^{k}\sum_{\bar{t}=0}^{p}\exp(-9(t+\bar{t})\log p)\\
&\leq p\sum_{t=1}^{k}\sum_{\bar{t}=0}^{p}\exp(-9\log p)\\
&\leq kp^2\frac{p+1}{p}p^{-9} = 2kp^{-7}.
\end{aligned}$$

□

Lemma 4.10. *One has*

$$\mathbb{P}\left(\bigcap_{j\in[p]}\bigcap_{S\subseteq[p]\setminus\{j\}}\mathcal{E}_4(j,S)\right)\geq 1-8kp^{-7}. \tag{4.99}$$

Proof. The proof of this lemma follows a similar path to the proof of Lemma 15 of [20]. Fix $j\in[p], S\subseteq[p]\setminus\{j\}$, and let $t=|S_j^*\setminus S|, \bar{t}=|S\setminus S_j^*|, S_j^0=S_j^*\setminus S$. Let $\mathcal{W}$ be the column span of $\boldsymbol{X}_{S\cap S_j^*}$. Moreover, let $\mathcal{U},\mathcal{V}$ be orthogonal complement of $\mathcal{W}$ as subspaces of column spans of $\boldsymbol{X}_S$ and $\boldsymbol{X}_{S_j^*}$, respectively. Let $\boldsymbol{P}_{\mathcal{U}},\boldsymbol{P}_{\mathcal{V}},\boldsymbol{P}_{\mathcal{W}}$ be projection matrices onto $\mathcal{U},\mathcal{V},\mathcal{W}$, respectively. With this notation in place, one has

$$\boldsymbol{\varepsilon}_j^\top(\boldsymbol{P}_{\boldsymbol{X}_S}-\boldsymbol{P}_{\boldsymbol{X}_{S_j^*}})\boldsymbol{\varepsilon}_j=\boldsymbol{\varepsilon}_j^\top(\boldsymbol{P}_{\mathcal{U}}-\boldsymbol{P}_{\mathcal{V}})\boldsymbol{\varepsilon}_j.$$

Note that $\dim(\mathcal{U})=\bar{t},\dim(\mathcal{V})=t$. As a result, by calculations similar to one in Lemma 15 of [20], we achieve

$$\mathbb{P}\left(\boldsymbol{\varepsilon}_j^\top\boldsymbol{P}_{\mathcal{U}}\boldsymbol{\varepsilon}_j\leq\bar{t}(\sigma_j^*)^2+(\sigma_j^*)^2x,\quad \boldsymbol{\varepsilon}_j^\top\boldsymbol{P}_{\mathcal{V}}\boldsymbol{\varepsilon}_j\geq-(\sigma_j^*)^2x\right)\geq \\ 1-2\exp(-c\min(x,x^2/t))-2\exp(-c\min(x,x^2/\bar{t})).$$

without loss of generality, we assume $\bar{t}\geq 1$ as otherwise the lemma is trivial. Take

$$x=\xi(t+\bar{t})\log p$$

for some sufficiently large absolute constant ξ and we achieve

$$\mathbb{P}\left(\boldsymbol{\varepsilon}_j^\top\boldsymbol{P}_{\mathcal{U}}\boldsymbol{\varepsilon}_j-\boldsymbol{\varepsilon}_j^\top\boldsymbol{P}_{\mathcal{V}}\boldsymbol{\varepsilon}_j\lesssim(\sigma_j^*)^2(t+\bar{t})\log p\right)\geq 1-4\exp(-10(t+\bar{t})\log p).$$

The proof is completed by union bound similar to Lemma 4.9. □

Lemma 4.11. *[20, Lemma 14] One has*

$$\mathbb{P}\left(\bigcap_{j\in[p]}\bigcap_{\substack{S\subseteq[p]\setminus\{j\}\\|S|\leq k}}\mathcal{E}_5(j,S)\right)\geq 1-2kp^{-7}. \tag{4.100}$$

Lemma 4.12. *Under the assumptions of Theorem 4.5,*

$$\mathbb{P}\left(\bigcap_{j\in[p]}\mathcal{E}_6(j)\right)\geq 1-12kp^{-7}. \tag{4.101}$$

Proof. In this proof, we assume without loss of generality that $|\hat{S}_j|\leq n$ as otherwise, it is possible to remove some redundant indices in $\hat{S}_j$ without increasing $\boldsymbol{\mathcal{L}}_{\hat{S}_j}(\boldsymbol{x}_j)$, as this quantity is zero in both cases. The proof of this lemma is on events $\mathcal{E}_1,\mathcal{E}_2,\mathcal{E}_3$ and $\mathcal{E}_4$ over all values of j,S, as in Lemmas 4.8, 4.9 and 4.10. The intersection of these events happen with probability at least

$$1-12kp^{-7}.$$

Recalling the definition of $\boldsymbol{\mathcal{L}}_S(\cdot)$ in (4.84), one has (see calculations leading to (89) of [20] and (6.1) of [82]),

$$n\boldsymbol{\mathcal{L}}_{S_j}(\boldsymbol{x}_j)=n(\boldsymbol{\beta}^*_{S_j^0,j})^\top(\hat{\boldsymbol{\Sigma}}/[S_j,S_j^*])\boldsymbol{\beta}^*_{S_j^0,j}+2\boldsymbol{\varepsilon}_j^\top(\boldsymbol{I}_n-\boldsymbol{P}_{\boldsymbol{X}_{S_j}})\boldsymbol{X}_{S_j^0}\boldsymbol{\beta}^*_{S_j^0,j}+\boldsymbol{\varepsilon}_j^\top(\boldsymbol{I}_n-\boldsymbol{P}_{\boldsymbol{X}_{S_j}})\boldsymbol{\varepsilon}_j \tag{4.102}$$

where $S_j^0 = S_j^* \setminus S_j$. As a result, one has

$$
\begin{aligned}
& n\left[\mathcal{L}_{\hat{S}_j}(\boldsymbol{x}_j) - \mathcal{L}_{S_j^*}(\boldsymbol{x}_j)\right] \\
\overset{(a)}{=}\ & n(\boldsymbol{\beta}^*_{S_j^0,j})^\top(\hat{\boldsymbol{\Sigma}}/[S_j, S_j^*])\boldsymbol{\beta}^*_{S_j^0,j} + 2\boldsymbol{\varepsilon}_j^\top(\boldsymbol{I}_n - \boldsymbol{P}_{\boldsymbol{X}_{S_j}})\boldsymbol{X}_{S_j^0}\boldsymbol{\beta}^*_{S_j^0,j} + \boldsymbol{\varepsilon}_j^\top(\boldsymbol{P}_{\boldsymbol{X}_{S_j^*}} - \boldsymbol{P}_{\boldsymbol{X}_{S_j}})\boldsymbol{\varepsilon}_j \\
\overset{(b)}{\geq}\ & n(\boldsymbol{\beta}^*_{S_j^0,j})^\top(\hat{\boldsymbol{\Sigma}}/[S_j, S_j^*])\boldsymbol{\beta}^*_{S_j^0,j} - 2c_{t_1}(\sigma_j^*)\sqrt{n(\boldsymbol{\beta}^*_{S_j^0,j})^\top(\hat{\boldsymbol{\Sigma}}/[S, S_j^*])\boldsymbol{\beta}^*_{S_j^0,j}}\sqrt{(t_j + \bar{t}_j)\log p} \\
& - c_{t_2}(\sigma_j^*)^2(t_j + \bar{t}_j)\log p \\
\overset{(c)}{\geq}\ & \frac{3}{4}n(\boldsymbol{\beta}^*_{S_j^0,j})^\top(\hat{\boldsymbol{\Sigma}}/[S_j, S_j^*])\boldsymbol{\beta}^*_{S_j^0,j} - 4c_{t_1}^2(\sigma_j^*)^2(t_j + \bar{t}_j)\log p - c_{t_2}(\sigma_j^*)^2(t_j + \bar{t}_j)\log p
\end{aligned}
\tag{4.103}
$$

where (a) is due to (4.102), (b) is due to events $\mathcal{E}_3, \mathcal{E}_4$ and (c) is by inequality $2ab \geq -a^2/4 - 4b^2$. Next, let $\tilde{S}_j = S_j \cap S_j^*$. Note that $S_j^0 = S_j \setminus \tilde{S}_j$. Write

$$
\begin{aligned}
& (\boldsymbol{\beta}^*_{S_j^0,j})^\top(\hat{\boldsymbol{\Sigma}}/[S_j, S_j^*])\boldsymbol{\beta}^*_{S_j^0,j} \\
=\ & (\boldsymbol{\beta}^*_{S_j^0,j})^\top(\hat{\boldsymbol{\Sigma}}/[S_j, S_j^*])\boldsymbol{\beta}^*_{S_j^0,j} - (\boldsymbol{\beta}^*_{S_j^0,j})^\top(\hat{\boldsymbol{\Sigma}}/[\tilde{S}_j, S_j^*])\boldsymbol{\beta}^*_{S_j^0,j} + (\boldsymbol{\beta}^*_{S_j^0,j})^\top(\hat{\boldsymbol{\Sigma}}/[\tilde{S}_j, S_j^*])\boldsymbol{\beta}^*_{S_j^0,j} \\
\overset{(a)}{=}\ & (\boldsymbol{\beta}^*_{S_j^0,j})^\top\left(\hat{\boldsymbol{\Sigma}}_{S_0^j,\tilde{S}_j}\hat{\boldsymbol{\Sigma}}^{-1}_{\tilde{S}_j,\tilde{S}_j}\hat{\boldsymbol{\Sigma}}_{\tilde{S}_j,S_j^0} - \hat{\boldsymbol{\Sigma}}_{S_0^j,S_j}\hat{\boldsymbol{\Sigma}}^{-1}_{S_j,S_j}\hat{\boldsymbol{\Sigma}}_{S_j,S_j^0}\right)(\boldsymbol{\beta}^*_{S_j^0,j}) \\
& + (\boldsymbol{\beta}^*_{S_j^0,j})^\top(\hat{\boldsymbol{\Sigma}}/[\tilde{S}_j, S_j^*])\boldsymbol{\beta}^*_{S_j^0,j} \\
\overset{(b)}{=}\ & \frac{1}{n}(\boldsymbol{X}_{S_j^0}\boldsymbol{\beta}^*_{S_j^0,j})^\top\left(\boldsymbol{X}_{\tilde{S}_j}(\boldsymbol{X}_{\tilde{S}_j}^\top\boldsymbol{X}_{\tilde{S}_j})^{-1}\boldsymbol{X}_{\tilde{S}_j}^\top - \boldsymbol{X}_{S_j}(\boldsymbol{X}_{S_j}^\top\boldsymbol{X}_{S_j})^{-1}\boldsymbol{X}_{S_j}^\top\right)(\boldsymbol{X}_{S_j^0}\boldsymbol{\beta}^*_{S_j^0,j}) \\
& + (\boldsymbol{\beta}^*_{S_j^0,j})^\top(\hat{\boldsymbol{\Sigma}}/[\tilde{S}_j, S_j^*])\boldsymbol{\beta}^*_{S_j^0,j} \\
\overset{(c)}{=}\ & \frac{1}{n}(\boldsymbol{X}_{S_j^0}\boldsymbol{\beta}^*_{S_j^0,j})^\top\left(\boldsymbol{P}_{\boldsymbol{X}_{\tilde{S}_j}} - \boldsymbol{P}_{\boldsymbol{X}_{S_j}}\right)(\boldsymbol{X}_{S_j^0}\boldsymbol{\beta}^*_{S_j^0,j}) + (\boldsymbol{\beta}^*_{S_j^0,j})^\top(\hat{\boldsymbol{\Sigma}}/[\tilde{S}_j, S_j^*])\boldsymbol{\beta}^*_{S_j^0,j} \\
\overset{(d)}{\geq}\ & -\frac{1}{n}(\boldsymbol{X}_{S_j^0}\boldsymbol{\beta}^*_{S_j^0,j})^\top\boldsymbol{P}_{\boldsymbol{X}_{S_j}}(\boldsymbol{X}_{S_j^0}\boldsymbol{\beta}^*_{S_j^0,j}) + (\boldsymbol{\beta}^*_{S_j^0,j})^\top(\hat{\boldsymbol{\Sigma}}/[\tilde{S}_j, S_j^*])\boldsymbol{\beta}^*_{S_j^0,j} \\
\overset{(e)}{\geq}\ & -\frac{1}{n}\left\|\boldsymbol{X}_{S_j^0}\boldsymbol{\beta}^*_{S_j^0,j}\right\|_2^2 + (\boldsymbol{\beta}^*_{S_j^0,j})^\top(\hat{\boldsymbol{\Sigma}}/[\tilde{S}_j, S_j^*])\boldsymbol{\beta}^*_{S_j^0,j} \\
\overset{(f)}{\geq}\ & 0.2\eta\frac{|\tilde{S}_j^0|\log p}{n} - 4\frac{|S_j^0|}{k}
\end{aligned}
\tag{4.104}
$$

where (a) is achieved by substituting the Schur complement definition, (b) is achieved by substituting $\hat{\boldsymbol{\Sigma}}$, (c) is by definition of projection matrices, (d) is true as projection matrices positive semidefinite, (e) is true as the largest eigenvalue of a projection ma-

trix is 1, and (f) is due to events $\mathcal{E}_1, \mathcal{E}_2$ as $|\tilde{S}_j| \leq k$. Substituting (4.104) into (4.103) completes the proof.

□

Lemma 4.13. *One has*

$$\mathbb{P}\left(\bigcap_{j\in[p]} \mathcal{E}_7(j)\right) \geq 1 - 2kp^{-7} - p(k/p)^{10}. \tag{4.105}$$

Proof. The proof of this lemma is on the event considered in (4.64) and the intersection of events $\mathcal{E}_5(j, S)$ for all j, S as in Lemma 4.11. Note that by union bound, this happens with probability greater than

$$1 - 2kp^{-7} - p(k/p)^{10}.$$

Based on the event considered in (4.64) (and arguments leading to (4.65)),

$$5(\sigma_j^*)^2 n/6 \leq \|\boldsymbol{\varepsilon}_j\|_2^2 \leq 7(\sigma_j^*)^2 n/6. \tag{4.106}$$

In addition, from (4.102),

$$n\mathcal{L}_{S_j^*}(\boldsymbol{x}_j) = \boldsymbol{\varepsilon}_j^\top (\boldsymbol{I}_n - \boldsymbol{P}_{\boldsymbol{X}_{S_j^*}})\boldsymbol{\varepsilon}_j = \|\boldsymbol{\varepsilon}_j\|_2^2 - \boldsymbol{\varepsilon}_j^\top \boldsymbol{P}_{\boldsymbol{X}_{S_j^*}} \boldsymbol{\varepsilon}_j. \tag{4.107}$$

As a result, from (4.106) we have

$$\frac{5(\sigma_j^*)^2}{6} - \frac{1}{n}\boldsymbol{\varepsilon}_j^\top \boldsymbol{P}_{\boldsymbol{X}_{S_j^*}} \boldsymbol{\varepsilon}_j \leq \mathcal{L}_{S_j^*}(\boldsymbol{x}_j) \leq \frac{7(\sigma_j^*)^2}{6} - \frac{1}{n}\boldsymbol{\varepsilon}_j^\top \boldsymbol{P}_{\boldsymbol{X}_{S_j^*}} \boldsymbol{\varepsilon}_j. \tag{4.108}$$

Moreover, by taking $n = c_n k \log p$ to be sufficiently large and by event $\mathcal{E}_4$,

$$-(\sigma_j^*)^2/6 \leq \frac{1}{n}\boldsymbol{\varepsilon}_j^\top \boldsymbol{P}_{\boldsymbol{X}_{S_j^*}} \boldsymbol{\varepsilon}_j \leq (\sigma_j^*)^2/6$$

which together with (4.108) completes the proof. □

Lemma 4.14. *Under the assumptions of Theorem 4.5, one has*

$$\mathbb{P}\left(\bigcap_{j\in\mathcal{J}}\left\{\frac{-99}{100}\leq\frac{\mathcal{L}_{\hat{S}_j}(\boldsymbol{x}_j)-\mathcal{L}_{S_j^*}(\boldsymbol{x}_j)}{\mathcal{L}_{S_j^*}(\boldsymbol{x}_j)}\leq 100\right\}\right)\geq 1-2p(k/p)^{10}-13kp^{-7}. \quad (4.109)$$

Proof. Note that $\boldsymbol{x}_j\sim\mathcal{N}(\boldsymbol{0},(\boldsymbol{\Sigma}^*)_{jj}\boldsymbol{I}_n)$. Let events A_j for $j\in[p]$ be defined as

$$A_j=\left\{\frac{1}{n}\|\boldsymbol{x}_j\|_2^2\leq\frac{7}{6}(\boldsymbol{\Sigma}^*)_{jj}\right\}. \quad (4.110)$$

By Lemma 4.5 and an argument similar to the one leading to (4.64), by taking $n\gtrsim\log p$, we have

$$\mathbb{P}(A_j)\geq 1-(k/p)^{10}$$

which leads to

$$\mathbb{P}\left(\bigcap_{j\in[p]}A_j\right)\geq 1-p(k/p)^{10}$$

by union bound. The proof of this lemma is on events $\bigcap_{j\in[p]}A_j$, and $\mathcal{E}_7$ over all choices of j, as considered in Lemma 4.13. By union bound, the intersection of these events occur with probability at least

$$1-2p(k/p)^{10}-2kp^{-7}.$$

First, for $j\in\mathcal{J}$ we have

$$\begin{aligned}\frac{\mathcal{L}_{\hat{S}_j}(\boldsymbol{x}_j)-\mathcal{L}_{S_j^*}(\boldsymbol{x}_j)}{\mathcal{L}_{S_j^*}(\boldsymbol{x}_j)}&\overset{(a)}{\geq}\frac{\ell}{\mathcal{L}_{S_j^*}(\boldsymbol{x}_j)}-1\\&\overset{(b)}{\geq}\frac{3\ell}{4(\sigma_j^*)^2}-1\\&\overset{(c)}{\geq}\frac{l_\sigma^2}{4u_\sigma^2}-1\geq-\frac{99}{100}\end{aligned} \quad (4.111)$$

where (a) is true as for $j\in\mathcal{J}$, $\mathcal{L}_{\hat{S}_j}(\boldsymbol{x}_j)\geq\ell$, (b) is due to event $\mathcal{E}_7$, (c) and the last inequality are due to Assumption 4.2 part **(B1)**. The proof of lower bound is

completed. Next, note that

$$\begin{aligned}\frac{\mathcal{L}_{\hat{S}_j}(\boldsymbol{x}_j) - \mathcal{L}_{S_j^*}(\boldsymbol{x}_j)}{\mathcal{L}_{S_j^*}(\boldsymbol{x}_j)} &\leq \frac{\mathcal{L}_{\hat{S}_j}(\boldsymbol{x}_j)}{\mathcal{L}_{S_j^*}(\boldsymbol{x}_j)} \\ &\overset{(a)}{\leq} \frac{\frac{1}{n}\|\boldsymbol{x}_j\|_2^2}{\mathcal{L}_{S_j^*}(\boldsymbol{x}_j)} \\ &\overset{(b)}{\leq} \frac{3}{2}\frac{\frac{1}{n}\|\boldsymbol{x}_j\|_2^2}{(\sigma_j^*)^2} \\ &\overset{(c)}{\leq} \frac{7}{4}\frac{(\boldsymbol{\Sigma}^*)_{jj}}{(\sigma_j^*)^2} \leq 100 \end{aligned} \tag{4.112}$$

where (a) is due to definition of $\mathcal{L}_{\hat{S}_j}$, (b) is by event $\mathcal{E}_7$, (c) is a result of event A_j and the last inequality is a result of Assumption 4.2 part **(B3)**. □

Lemma 4.15. *Let h_j be defined as in* (4.88)*. Let the event $\mathcal{E}_{\mathcal{J}^c}$ be defined as*

$$\begin{aligned}\mathcal{E}_{\mathcal{J}^c} = \Bigg\{ \sum_{j\in\mathcal{J}^c} \Big[h_j(\hat{\sigma}_j, \hat{S}_j) + \lambda|\hat{S}_j| - h_j(\tilde{\sigma}_j, S^*) - \lambda|S_j^*|\Big] \geq \\ \frac{c_1}{l_\sigma^2}\eta\frac{\log p}{n}\sum_{j\in\mathcal{J}^c}\tilde{t}_j + (c_\lambda - c_2)\frac{\log p}{n}\sum_{j\in\mathcal{J}^c}\bar{t}_j + \left(-c_\lambda - c_3 - \frac{c_n c_4}{l_\sigma^2}\right)\frac{\log p}{n}\sum_{j\in\mathcal{J}^c} t_j\Bigg\}\end{aligned} \tag{4.113}$$

for some absolute constants $c_1, \cdots, c_4 > 0$. Then,

$$\mathbb{P}(\mathcal{E}_{\mathcal{J}^c}) \geq 1 - 14kp^{-7} - p(k/p)^{10}.$$

Proof. The proof of this lemma in on the intersection of events $\mathcal{E}_6$ and $\mathcal{E}_7$ as in Lemmas 4.12 and 4.13. Note that this happens with probability at least

$$1 - 14kp^{-7} - p(k/p)^{10}.$$

One has

$$
\begin{aligned}
& h_j(\hat{\sigma}_j, \tilde{S}_j) - h_j(\sigma_j^*, S_j^*) \\
&= \log(\hat{\sigma}_j) + \frac{\mathcal{L}_{\hat{S}_j}(\boldsymbol{x}_j)}{2\hat{\sigma}_j^2} - \log(\tilde{\sigma}_j) - \frac{\mathcal{L}_{S_j^*}(\boldsymbol{x}_j)}{2\tilde{\sigma}_j^2} \\
&= \left[\log(\hat{\sigma}_j) - \log(\tilde{\sigma}_j) + \mathcal{L}_{S_j^*}(\boldsymbol{x}_j)(\frac{1}{2\hat{\sigma}_j} - \frac{1}{2\tilde{\sigma}_j})\right] + \frac{\mathcal{L}_{\hat{S}_j}(\boldsymbol{x}_j) - \mathcal{L}_{S_j^*}(\boldsymbol{x}_j)}{2\hat{\sigma}_j} \\
&\overset{(a)}{\geq} \frac{\mathcal{L}_{\hat{S}_j}(\boldsymbol{x}_j) - \mathcal{L}_{S_j^*}(\boldsymbol{x}_j)}{2\hat{\sigma}_j} \\
&\overset{(b)}{=} \frac{\mathcal{L}_{\hat{S}_j}(\boldsymbol{x}_j) - \mathcal{L}_{S_j^*}(\boldsymbol{x}_j)}{2\ell} \\
&\overset{(c)}{\geq} \frac{\frac{3}{20}\eta\tilde{t}_j\frac{\log p}{n} - (\sigma_j^*)^2\frac{(t_j+\bar{t}_j)\log p}{n}(4c_{t_1}^2 + c_{t_2}) - \frac{4t_j}{k}}{2\ell} \\
&\overset{(d)}{\geq} \frac{9}{40l_\sigma^2}\eta\tilde{t}_j\frac{\log p}{n} - \frac{6t_j}{l_\sigma^2 k} - \frac{3u_\sigma^2}{2l_\sigma^2}(4c_{t_1}^2 + c_{t_2})\frac{\log p}{n}t_j - \frac{3u_\sigma^2}{2l_\sigma^2}(4c_{t_1}^2 + c_{t_2})\frac{\log p}{n}\bar{t}_j \\
&\overset{(e)}{\geq} \frac{9}{40l_\sigma^2}\eta\tilde{t}_j\frac{\log p}{n} - \frac{6c_n}{l_\sigma^2}\frac{\log p}{n}t_j - \frac{75}{2}(4c_{t_1}^2 + c_{t_2})\frac{\log p}{n}t_j - \frac{75}{2}(4c_{t_1}^2 + c_{t_2})\frac{\log p}{n}\bar{t}_j \quad (4.114)
\end{aligned}
$$

where (a) is true as on event $\mathcal{E}_7$, $\mathcal{L}_{S_j^*}(\boldsymbol{x}_j) \geq 2(\sigma_j^*)^2/3 > \ell$ so $\tilde{\sigma}_j = \sqrt{\mathcal{L}_{S_j^*}(\boldsymbol{x}_j)}$ and $h_j(\tilde{\sigma}_j, S_j^*) \leq h_j(\hat{\sigma}_j, S_j^*)$, (b) is true as $\hat{\sigma}_j \geq \ell$, (c) is by event $\mathcal{E}_6$, (d) is by substituting $\ell = l_\sigma^2/3$ and $\sigma_j^* \leq u_\sigma$, and (e) is by Assumption 4.2 part **(B1)**, $u_\sigma/l_\sigma \leq 5$ and also $n = c_n k \log p$. By summing (4.114) over $j \in \mathcal{J}^c$, we achieve

$$
\begin{aligned}
&\sum_{j\in\mathcal{J}^c}\left[h_j(\hat{\sigma}_j, \hat{S}_j) + \lambda|\hat{S}_j| - h_j(\tilde{\sigma}_j, S^*) - \lambda|S_j^*|\right] \geq \\
&\qquad \frac{c_1}{l_\sigma^2}\eta\frac{\log p}{n}\sum_{j\in\mathcal{J}^c}\tilde{t}_j + (c_\lambda - c_2)\frac{\log p}{n}\sum_{j\in\mathcal{J}^c}\bar{t}_j + \left(-c_\lambda - c_3 - \frac{c_n c_4}{l_\sigma^2}\right)\frac{\log p}{n}\sum_{j\in\mathcal{J}^c}t_j
\end{aligned}
$$

$c_1 = 9/40$, $c_2 = c_3 = 75(4c_{t_1}^2 + c_{t_2})/2$ and $c_4 = 6$. □

Lemma 4.16. *Let $a > 0$. Then,*

$$\log(1+x) \geq \frac{x}{1+a}$$

for $x \in [0, a]$. *Similarly, if* $a \in (-1, 0)$,

$$\log(1+x) \geq \frac{x}{1+a}$$

for $x \in [a, 0]$.

Proof. Suppose $a > 0$ and $x \in [0, a]$. Note that

$$\log(1+x) = \int_0^x \frac{dt}{1+t} \geq \int_0^x \frac{dt}{1+a} = \frac{x}{1+a}.$$

The proof of other part is similar. □

Lemma 4.17. *Let the event* $\mathcal{E}_{\mathcal{J}}$ *be defined as*

$$\mathcal{E}_{\mathcal{J}} = \Bigg\{ \sum_{j \in \mathcal{J}} \Big[h_j(\hat{\sigma}_j, \hat{S}_j) + \lambda |\hat{S}_j| - h_j(\tilde{\sigma}_j, S^*) - \lambda |S_j^*| \Big] \geq \\ \frac{c_5 \eta}{u_\sigma^2} \frac{\log p}{n} \sum_{j \in \mathcal{J}} \tilde{t}_j + (-c_6 - \frac{c_7 c_n}{l_\sigma^2} - c_\lambda) \frac{\log p}{n} \sum_{j \in \mathcal{J}} t_j + (c_\lambda - c_8) \frac{\log p}{n} \sum_{j \in \mathcal{J}} \bar{t}_j \Bigg\} \quad (4.115)$$

for some absolute constants $c_5, \cdots, c_8 > 0$. *Then,*

$$\mathbb{P}(\mathcal{E}_{\mathcal{J}}) \geq 1 - 27kp^{-7} - 3p(k/p)^{10}.$$

Proof. The proof of this lemma in on the intersection of events $\mathcal{E}_6$, $\mathcal{E}_7$ as in Lemmas 4.12 and 4.13 and the event in Lemma 4.14. Note that this happens with probability at least

$$1 - 27kp^{-7} - 3p(k/p)^{10}.$$

Let $\mathcal{J}_+, \mathcal{J}_- \subseteq [p]$ be defined as

$$\begin{aligned} \mathcal{J}_+ &= \{ j \in \mathcal{J} : \mathcal{L}_{\hat{S}_j}(\boldsymbol{x}_j) - \mathcal{L}_{S_j^*}(\boldsymbol{x}_j) \geq 0 \} \\ \mathcal{J}_- &= \{ j \in \mathcal{J} : \mathcal{L}_{\hat{S}_j}(\boldsymbol{x}_j) - \mathcal{L}_{S_j^*}(\boldsymbol{x}_j) < 0 \}. \end{aligned} \quad (4.116)$$

Based on Lemma 4.14, for $j \in \mathcal{J}_+$, we have

$$0 \leq \frac{\mathcal{L}_{\hat{S}_j}(\boldsymbol{x}_j) - \mathcal{L}_{S_j^*}(\boldsymbol{x}_j)}{\mathcal{L}_{S_j^*}(\boldsymbol{x}_j)} \leq 100.$$

Consequently, by Lemma 4.16, for $j \in \mathcal{J}_+$ we have

$$\log\left(1 + \frac{\mathcal{L}_{\hat{S}_j}(\boldsymbol{x}_j) - \mathcal{L}_{S_j^*}(\boldsymbol{x}_j)}{\mathcal{L}_{S_j^*}(\boldsymbol{x}_j)}\right) \geq c^{(1)} \frac{\mathcal{L}_{\hat{S}_j}(\boldsymbol{x}_j) - \mathcal{L}_{S_j^*}(\boldsymbol{x}_j)}{\mathcal{L}_{S_j^*}(\boldsymbol{x}_j)} \tag{4.117}$$

where $c^{(1)} = 1/101$. Similarly, for $j \in \mathcal{J}_-$ we have

$$\log\left(1 + \frac{\mathcal{L}_{\hat{S}_j}(\boldsymbol{x}_j) - \mathcal{L}_{S_j^*}(\boldsymbol{x}_j)}{\mathcal{L}_{S_j^*}(\boldsymbol{x}_j)}\right) \geq c^{(2)} \frac{\mathcal{L}_{\hat{S}_j}(\boldsymbol{x}_j) - \mathcal{L}_{S_j^*}(\boldsymbol{x}_j)}{\mathcal{L}_{S_j^*}(\boldsymbol{x}_j)} \tag{4.118}$$

where $c^{(2)} = 100$. By discussion leading to (4.90), we have that for $j \in \mathcal{J}$,

$$h_j(\hat{\sigma}_j, \hat{S}_j) = \frac{\log(\mathcal{L}_{\hat{S}_j}(\boldsymbol{x}_j))}{2} + \frac{1}{2}, h_j(\tilde{\sigma}_j, S_j^*) = \frac{\log(\mathcal{L}_{S_j^*}(\boldsymbol{x}_j))}{2} + \frac{1}{2}.$$

Therefore, one has

$$
\begin{aligned}
&\sum_{j\in\mathcal{J}}\left\{h_j(\hat{\sigma}_j,\hat{S}_j)-h_j(\tilde{\sigma}_j,S_j^*)+\lambda|\hat{S}_j|-\lambda|S_j^*|\right\}\\
=&\sum_{j\in\mathcal{J}}\left\{\frac{1}{2}\log(\mathcal{L}_{\hat{S}_j}(\boldsymbol{x}_j))-\frac{1}{2}\log(\mathcal{L}_{S_j^*}(\boldsymbol{x}_j))+\lambda|\hat{S}_j|-\lambda|S_j^*|\right\}\\
=&\sum_{j\in\mathcal{J}}\frac{1}{2}\log\left(1+\frac{\mathcal{L}_{\hat{S}_j}(\boldsymbol{x}_j)-\mathcal{L}_{S_j^*}(\boldsymbol{x}_j)}{\mathcal{L}_{S_j^*}(\boldsymbol{x}_j)}\right)+\lambda\sum_{j\in\mathcal{J}}(\bar{t}_j-t_j)\\
\overset{(a)}{=}&\sum_{j\in\mathcal{J}_+}\frac{1}{2}\log\left(1+\frac{\mathcal{L}_{\hat{S}_j}(\boldsymbol{x}_j)-\mathcal{L}_{S_j^*}(\boldsymbol{x}_j)}{\mathcal{L}_{S_j^*}(\boldsymbol{x}_j)}\right)\\
&\qquad+\sum_{j\in\mathcal{J}_-}\frac{1}{2}\log\left(1+\frac{\mathcal{L}_{\hat{S}_j}(\boldsymbol{x}_j)-\mathcal{L}_{S_j^*}(\boldsymbol{x}_j)}{\mathcal{L}_{S_j^*}(\boldsymbol{x}_j)}\right)+\lambda\sum_{j\in\mathcal{J}}(\bar{t}_j-t_j)\\
\overset{(b)}{\geq}&c^{(1)}\sum_{j\in\mathcal{J}_+}\frac{\mathcal{L}_{\hat{S}_j}(\boldsymbol{x}_j)-\mathcal{L}_{S_j^*}(\boldsymbol{x}_j)}{\mathcal{L}_{S_j^*}(\boldsymbol{x}_j)}+c^{(2)}\sum_{j\in\mathcal{J}_-}\frac{\mathcal{L}_{\hat{S}_j}(\boldsymbol{x}_j)-\mathcal{L}_{S_j^*}(\boldsymbol{x}_j)}{\mathcal{L}_{S_j^*}(\boldsymbol{x}_j)}+\lambda\sum_{j\in\mathcal{J}}(\bar{t}_j-t_j)\\
\overset{(c)}{\geq}&\frac{3c^{(1)}}{4}\sum_{j\in\mathcal{J}_+}\frac{\mathcal{L}_{\hat{S}_j}(\boldsymbol{x}_j)-\mathcal{L}_{S_j^*}(\boldsymbol{x}_j)}{(\sigma_j^*)^2}+\frac{3c^{(2)}}{2}\sum_{j\in\mathcal{J}_-}\frac{\mathcal{L}_{\hat{S}_j}(\boldsymbol{x}_j)-\mathcal{L}_{S_j^*}(\boldsymbol{x}_j)}{(\sigma_j^*)^2}+\lambda\sum_{j\in\mathcal{J}}(\bar{t}_j-t_j)
\end{aligned}
\tag{4.119}
$$

where (a) is by the fact that $\mathcal{J}_+,\mathcal{J}_-$ is a partition of $\mathcal{J}$, (b) is due to (4.117) and (4.118), and (c) is due to event $\mathcal{E}_7$. From (4.119) and event $\mathcal{E}_6$,

$$
\begin{aligned}
&\sum_{j\in\mathcal{J}}\left\{h_j(\hat{\sigma}_j,\hat{S}_j)-h_j(\tilde{\sigma}_j,S_j^*)+\lambda|\hat{S}_j|-\lambda|S_j^*|\right\}\\
\geq&\frac{3c^{(1)}}{4}\sum_{j\in\mathcal{J}_+}\frac{\frac{3}{20}\eta\frac{\tilde{t}_j\log p}{n}-(\sigma_j^*)^2\frac{(t_j+\bar{t}_j)\log p}{n}(4c_{t_1}^2+c_{t_2})-\frac{4t_j}{k}}{(\sigma_j^*)^2}\\
&+\frac{3c^{(2)}}{2}\sum_{j\in\mathcal{J}_-}\frac{\frac{3}{20}\eta\frac{\tilde{t}_j\log p}{n}-(\sigma_j^*)^2\frac{(t_j+\bar{t}_j)\log p}{n}(4c_{t_1}^2+c_{t_2})-\frac{4t_j}{k}}{(\sigma_j^*)^2}+c_\lambda\frac{\log p}{n}\sum_{j\in\mathcal{J}}(\bar{t}_j-t_j)\\
\overset{(a)}{\geq}&\sum_{j\in\mathcal{J}}\Bigg[\frac{9c^{(1)}\eta}{80}\frac{\log p}{n}\tilde{t}_j-\frac{3c^{(2)}(4c_{t_1}^2+c_{t_2})}{2}\frac{\log p}{n}t_j-\frac{6c^{(2)}}{l_\sigma^2}c_n\frac{\log p}{n}t_j\\
&\qquad\qquad-c_\lambda t_j-\frac{3c^{(2)}(4c_{t_1}^2+c_{t_2})}{2}\frac{\log p}{n}\bar{t}_j+c_\lambda\bar{t}_j\Bigg]\\
\geq&\frac{c_5\eta}{u_\sigma^2}\frac{\log p}{n}\sum_{j\in\mathcal{J}}\tilde{t}_j+(-c_6-\frac{c_7c_n}{l_\sigma^2}-c_\lambda)\frac{\log p}{n}\sum_{j\in\mathcal{J}}t_j+(c_\lambda-c_8)\frac{\log p}{n}\sum_{j\in\mathcal{J}}\bar{t}_j
\end{aligned}
\tag{4.120}
$$

where (a) is due to the fact $c^{(2)} > c^{(1)}$, and $c_5 = 9c^{(1)}/80$, $c_6 = c_8 = 3c^{(2)}(4c_{t_1}^2 + c_{t_2})/2$ and $c_7 = 6c^{(2)}$. □

Proof of Theorem 4.5. The proof of this theorem is on the intersection of events $\mathcal{E}_{\mathcal{J}}$ and $\mathcal{E}_{\mathcal{J}^c}$ as in Lemmas 4.17 and 4.15. Note that this happens with probability at least

$$1 - 4p(k/p)^{10} - 41kp^{-7}. \tag{4.121}$$

By optimality of $\hat{z}$ and feasibility of z^*, we have

$$\begin{aligned}
0 \geq & \sum_{j=1}^{p} \Big\{ h_j(\hat{\sigma}_j, \hat{S}_j) - h_j(\tilde{\sigma}_j, S_j^*) + \lambda|\hat{S}_j| - \lambda|S_j^*| \Big\} \\
= & \sum_{j\in\mathcal{J}} \Big\{ h_j(\hat{\sigma}_j, \hat{S}_j) - h_j(\tilde{\sigma}_j, S_j^*) + \lambda|\hat{S}_j| - \lambda|S_j^*| \Big\} \\
& \qquad + \sum_{j\in\mathcal{J}^c} \Big\{ h_j(\hat{\sigma}_j, \hat{S}_j) - h_j(\tilde{\sigma}_j, S_j^*) + \lambda|\hat{S}_j| - \lambda|S_j^*| \Big\} \\
\overset{(a)}{\geq} & \frac{c_5\eta}{u_\sigma^2}\frac{\log p}{n}\sum_{j\in\mathcal{J}}\tilde{t}_j + (-c_6 - \frac{c_7 c_n}{l_\sigma^2} - c_\lambda)\frac{\log p}{n}\sum_{j\in\mathcal{J}} t_j + (c_\lambda - c_8)\frac{\log p}{n}\sum_{j\in\mathcal{J}}\bar{t}_j \\
& + \frac{c_1}{l_\sigma^2}\eta\frac{\log p}{n}\sum_{j\in\mathcal{J}^c}\tilde{t}_j + (c_\lambda - c_2)\frac{\log p}{n}\sum_{j\in\mathcal{J}^c}\bar{t}_j + \left(-c_\lambda - c_3 - \frac{c_n c_4}{l_\sigma^2}\right)\frac{\log p}{n}\sum_{j\in\mathcal{J}^c} t_j \\
\geq & \frac{c_{f_1}\eta}{u_\sigma^2}\frac{\log p}{n}\sum_{j=1}^{p}\tilde{t}_j + \left(-c_\lambda - c_{f_2} - \frac{c_n c_{f_3}}{l_\sigma^2}\right)\frac{\log p}{n}\sum_{j=1}^{p} t_j + (c_\lambda - c_{f_4})\frac{\log p}{n}\sum_{j=1}^{p}\bar{t}_j \\
\overset{(b)}{=} & \left[\frac{c_{f_1}\eta}{u_\sigma^2} - 2c_\lambda - 2c_{f_2} - \frac{2c_n c_{f_3}}{l_\sigma^2}\right]\frac{\log p}{n}\sum_{j=1}^{p}\tilde{t}_j + (c_\lambda - c_{f_4})\frac{\log p}{n}\sum_{j=1}^{p}\bar{t}_j
\end{aligned} \tag{4.122}$$

where $c_{f_1} = c_1 \wedge c_5$, $c_{f_2} = c_3 \vee c_6$, $c_{f_3} = c_4 \vee c_7$ and $c_{f_4} = c_2 \vee c_8$, (a) is due to Lemmas 4.17 and 4.15 and (b) is true as if $\hat{z}_{ij} \neq z_{ij}^*$, then $\hat{z}_{ji} \neq z_{ji}^*$ so $\sum_{j=1}^p t_j = 2\sum_{j=1}^p \tilde{t}_j$. Take $c_\lambda > c_{f_4}$ and $\eta \gtrsim (2c_\lambda + 2c_{f_2} + \frac{2c_n c_{f_3}}{l_\sigma^2})u_\sigma^2$. Therefore, from (4.122) we have

$$0 \geq \underbrace{\left[\frac{c_{f_1}\eta}{u_\sigma^2} - 2c_\lambda - 2c_{f_2} - \frac{2c_n c_{f_3}}{l_\sigma^2}\right]\frac{\log p}{n}}_{>0}\sum_{j=1}^{p}\tilde{t}_j + \underbrace{(c_\lambda - c_{f_4})\frac{\log p}{n}}_{>0}\sum_{j=1}^{p}\bar{t}_j$$

which implies $\sum_{j=1}^p \tilde{t}_j = \sum_{j=1}^p \bar{t}_j = 0$ or equivalently $\hat{z}_{ij} = z_{ij}^*$.

□

4.C Details of Example 4.1

Note that based on the definition of $\mathbf{\Theta}^*$ in (4.22), for $i \in [n]$, $\mathbb{E}[(\boldsymbol{x}_1)_i^2] = 1/(1-c^2)$. As a result, the distribution of the random variable $\frac{(\boldsymbol{x}_1)_i^2}{1/(1-c^2)}$ is $\chi^2(1)$ so

$$\mathrm{var}((\boldsymbol{x}_1)_i^2) = \frac{2}{(1-c^2)^2}, \mathbb{E}[(\boldsymbol{x}_1)_i^4] = \frac{3}{(1-c^2)^2}. \tag{4.123}$$

It is also easy to see

$$\begin{aligned} &\mathbb{E}\left[\left((\boldsymbol{x}_1)_i^2(1-c^2)\right)\left((\boldsymbol{x}_2)_i^2(1-c^2)\right)\right] \\ &= 1 + 2\mathbb{E}\left[\left((\boldsymbol{x}_1)_i\sqrt{1-c^2}\right)\left((\boldsymbol{x}_2)_i\sqrt{1-c^2}\right)\right]^2 = 1 + 2c^2. \end{aligned} \tag{4.124}$$

Let $y_i = \frac{(\boldsymbol{x}_1)_i^2}{2} + \frac{(\boldsymbol{x}_2)_i^2}{2}$ for $i \in [n]$. Note that $\mathbb{E}[y_i] = 1/(1-c^2)$. As a result,

$$\begin{aligned} \mathrm{var}(y_i) &= \frac{1}{4}\mathbb{E}[((\boldsymbol{x}_1)_i^2 + (\boldsymbol{x}_2)_i^2)^2] - \mathbb{E}[y_i]^2 \\ &\overset{(a)}{=} \frac{3}{4(1-c^2)^2} + \frac{3}{4(1-c^2)^2} + \frac{\mathbb{E}[(\boldsymbol{x}_1)_i^2(\boldsymbol{x}_2)_i^2]}{2} - \frac{1}{(1-c^2)^2} \\ &= \frac{1 + \mathbb{E}[\left((\boldsymbol{x}_1)_i^2(1-c^2)\right)\left((\boldsymbol{x}_2)_i^2(1-c^2)\right)]}{2(1-c^2)^2} \\ &\overset{(b)}{=} \frac{2+2c^2}{2(1-c^2)^2} = \frac{1+c^2}{(1-c^2)^2}. \end{aligned} \tag{4.125}$$

where (a) is due to (4.123) and (b) is due to (4.124). In addition,

$$\begin{aligned} \mathrm{var}((\boldsymbol{x}_1)_i(\boldsymbol{x}_2)_i) &= \frac{\mathbb{E}[\left((\boldsymbol{x}_1)_i\sqrt{1-c^2}\right)^2\left((\boldsymbol{x}_2)_i\sqrt{1-c^2}\right)^2]}{(1-c^2)^2} - \frac{c^2}{(1-c^2)^2} \\ &\overset{(a)}{=} \frac{1+c^2}{(1-c^2)^2} \end{aligned} \tag{4.126}$$

where (a) is due to (4.124). Define the events

$$\begin{aligned}
\mathcal{E}_1 &= \left\{ \left| \frac{1}{n} \boldsymbol{x}_1^\top \boldsymbol{x}_2 + \frac{c}{1-c^2} \right| > \frac{\epsilon}{1-c^2} \right\}, \\
\mathcal{E}_2 &= \left\{ \left| \frac{1}{n} \boldsymbol{x}_1^\top \boldsymbol{x}_1 - \frac{1}{1-c^2} \right| > \frac{\epsilon}{1-c^2} \right\}, \\
\mathcal{E}_3 &= \left\{ \left| \frac{1}{n} \boldsymbol{x}_2^\top \boldsymbol{x}_2 - \frac{1}{1-c^2} \right| > \frac{\epsilon}{1-c^2} \right\}, \\
\mathcal{E}_4 &= \left\{ \left| \frac{1}{n} \sum_{i=1}^n y_i - \frac{1}{1-c^2} \right| > \frac{\epsilon}{1-c^2} \right\}.
\end{aligned} \tag{4.127}$$

By Chebyshev's inequality, (4.123), (4.125) and (4.126),

$$\mathbb{P}(\mathcal{E}_1^c) \le \frac{1+c^2}{n\epsilon^2},\ \mathbb{P}(\mathcal{E}_2^c) \le \frac{2}{n\epsilon^2},\ ,\mathbb{P}(\mathcal{E}_3^c) \le \frac{2}{n\epsilon^2},\ \mathbb{P}(\mathcal{E}_4^c) \le \frac{1+c^2}{n\epsilon^2}. \tag{4.128}$$

As a result, for the symmetric error (4.24), on events $\mathcal{E}_1, \mathcal{E}_4$ we have we have

$$\begin{aligned}
& 2\left(\beta^* - \frac{2\boldsymbol{x}_1^\top \boldsymbol{x}_2}{\boldsymbol{x}_1^\top \boldsymbol{x}_1 + \boldsymbol{x}_2^\top \boldsymbol{x}_2}\right)^2 \\
&= 2\left(c + \frac{\boldsymbol{x}_1^\top \boldsymbol{x}_2/n}{\sum_{i=1}^n y_i/n}\right)^2 \\
&= \frac{2}{(\sum_{i=1}^n y_i/n)^2}\left(c\sum_{i=1}^n y_i/n + \boldsymbol{x}_1^\top \boldsymbol{x}_2/n\right)^2 \\
&\le \frac{4(1-c^2)^2}{(1-\epsilon)^2}\left[(c\sum_{i=1}^n y_i/n - c/(1-c^2))^2 + (\boldsymbol{x}_1^\top \boldsymbol{x}_2/n + c/(1-c^2))^2\right] \\
&\le \frac{4(1-c^2)^2}{(1-\epsilon)^2}\left[\frac{c^2\epsilon^2}{(1-c^2)^2} + \frac{\epsilon^2}{(1-c^2)^2}\right] = \frac{4\epsilon^2(c^2+1)}{(1-\epsilon)^2}.
\end{aligned} \tag{4.129}$$

Similarly, for the asymmetric case (4.26), on events $\mathcal{E}_1, \mathcal{E}_2, \mathcal{E}_3$ we have

$$\begin{aligned}
&\left(\beta^* - \frac{\boldsymbol{x}_1^\top \boldsymbol{x}_2}{\boldsymbol{x}_1^\top \boldsymbol{x}_1}\right)^2 + \left(\beta^* - \frac{\boldsymbol{x}_1^\top \boldsymbol{x}_2}{\boldsymbol{x}_2^\top \boldsymbol{x}_2}\right)^2 \\
&= \frac{\left(c\boldsymbol{x}_1^\top \boldsymbol{x}_1/n + \boldsymbol{x}_1^\top \boldsymbol{x}_2/n\right)^2}{(\boldsymbol{x}_1^\top \boldsymbol{x}_1/n)^2} + \frac{\left(c\boldsymbol{x}_2^\top \boldsymbol{x}_2/n + \boldsymbol{x}_1^\top \boldsymbol{x}_2/n\right)^2}{(\boldsymbol{x}_2^\top \boldsymbol{x}_2/n)^2} \\
&\leq \frac{2(1-c^2)^2}{(1-\epsilon)^2}\frac{\epsilon^2(c^2+1)}{(1-c^2)^2} + \frac{2(1-c^2)^2}{(1-\epsilon)^2}\frac{\epsilon^2(c^2+1)}{(1-c^2)^2} \\
&= \frac{4\epsilon^2(c^2+1)}{(1-\epsilon)^2}.
\end{aligned} \tag{4.130}$$

Chapter 5

Safe Screening Procedure within a Specialized Branch-and-Bound Solver for Sparse Learning

It is a joint work with Xiang Meng and Rahul Mazumder.

5.1 Introduction

Sparse learning is a central problem in high-dimensional statistics, machine learning and other related fields [115]. We consider the sparse learning problem with $\ell_0\ell_2$ regularization [161, 116, 120]:

$$\min_{\boldsymbol{\beta}} \; F_0(\boldsymbol{\beta}) := f(\boldsymbol{\beta}) + \lambda_0\|\boldsymbol{\beta}\|_0 + \lambda_2\|\boldsymbol{\beta}\|^2, \tag{5.1}$$

where f is continuously differentiable convex function based on the data $(\boldsymbol{X}, \boldsymbol{y}) \in \mathbb{R}^{n\times p} \times \mathbb{R}^p$ and $\boldsymbol{\beta} \in \mathbb{R}^p$ is the vector of decision variables. Two important applications of Problem (5.1) are sparse linear regression with $f(\boldsymbol{\beta}) = \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|^2$ and sparse logistic regression with $f(\boldsymbol{\beta}) = \sum_{j=1}^n \log(1 + \exp(-y_j \boldsymbol{x}_j^\top \boldsymbol{\beta}))$.

In Problem (5.1), $\|\boldsymbol{\beta}\|_0$ denotes the number of nonzeros coordinates in $\boldsymbol{\beta}$, and $\|\boldsymbol{\beta}\|^2$ is the squared ℓ_2 norm of $\boldsymbol{\beta}$. Note that the inclusion of ℓ_2 term can help preventing

overfitting issues in the low-signal-to-noise-ratio (SNR) regimes [114, 161]. There have been extensive studies in statistical properties of ℓ_0-based estimators in the statistical literature [98, 43, 187, 46, 230, 65, 212, 161, 87, 67]. These estimators have been shown to exhibit some intriguing properties in support recovery, estimation error and prediction error under certain circumstances.

Nevertheless, Problem (5.1) is NP-Hard [169] and thus is computationally challenging. Thanks to the significant advancements in the field of mixed-integer optimization (MIO), there has been an emerging interest in developing MIO-based approaches to solve Problem (5.1) and its siblings, e.g. [58, 28, 167, 182, 72, 30, 116, 33, 222, 9, 120, 67, 165]. Specifically, several works [28, 97, 167, 182] have demonstrated that off-the-shelf MIP solvers can handle problem instances of moderate size ($p \lesssim 10^3$). While they can achieve promising results, they are still relatively slow for practical usage [114, 116], compared to specialized solvers tailored for its ℓ_1 counterparts and local heuristics for (5.1), e.g. [89, 114, 118, 222]. Recently, there have been several attempts to develop specialized MIO solvers for ℓ_0-based estimators [135, 30, 33, 67, 120, 165]. In particular, [120] develop a specialized nonlinear Branch-and-Bound (BnB) solver for Problem (5.1) in the context of linear regression — they leverage several strategies and heuristics to improve the efficiency of convex relaxation subproblem solving at each node within the BnB tree. They demonstrate a significant speedup, compared to commercial solvers (Gurobi [104], Mosek [6]) as well as state-of-the-art solver [30]. In an orthogonal direction, recently, [9] propose a safe screening approach to provably identify some of the zero and non-zero entries in the minimizers and thus reduce the number of binary variables, *prior to* solving Problem (5.1) for linear regression. However, in our experiments, we observe that this preprocessing screening might not lead to significant reduction in the problem size. In this chapter, we bring this notation of safe screening into the BnB framework to progressively screen variables for each node *during the course of* solving Problem (5.1).

To this end, we propose a nonlinear Branch-and-Bound framework with safe screenings at each node for solving Problem (5.1). Our framework can handle

any smooth convex function f, including the linear regression and logistic regression problems. We consider the hybrid perspective reformulation proposed by [120], which incorporates perspective formulation [88, 4, 101] and big-M reformulation [28, 222, 161, 116] as special cases.

In brief, the safe screening of our framework works as follows: at each node, after solving the node relaxation problem, the dual bounds for child nodes are computed to compare with the best feasible integral solution so as to determine the set of binary variables that can be safely fixed to 0 or 1 for all descendants of the current node. Such screening strategy can help improve the efficiency of the Branch-and-Bound solver in several ways: (a) it reduces the optimization complexity for relaxation subproblems in that more and more variables are fixed to 0 or 1 as the tree goes deeper; (b) it improves the branching efficiency by eliminating the unnecessary candidates for branching, especially for strong branching rules [5, 69] that involve additional relaxation problem solving. Additionally, combined with strong branching, we can further enhance the quality of screening based on the additional information brought by the strong branching, which further reduces the search space and thus reduces the size of tree (in term of tree depth and the number of nodes).

To better exploit the sparsity structure, the first-order primal methods are usually used for solving the relaxation subproblems, which might lead to an approximate solution, namely a primal feasible solution with low-to-medium accuracy. In contrast to [9, 70] which requires an optimal solution to the relaxation problem, our framework supports an approximate solution to perform the safe screening.

Contributions and Structure The key contributions are summarized as below.

- We consider the sparse learning problem (5.1) with a general convex smooth function f, and develop a nonlinear Branch-and-Bound solver for the hybrid perspective reformulation (See Section 5.2)
- We investigate into the optimality conditions and dual characterizations of convex relaxation subproblems, and present a dual bound formulation for any given primal feasible solution (see Section 5.3).

- We propose a safe screening framework at each node within the Branch-and-Bound tree, and show how this interacts with different branching strategies to improve the efficiency of BnB. For strong branching rules, we propose an enhanced version which leads to further improvement (see Section 5.4).

- Our BnB framework can handle various losses in sparse learning, including squared error loss and Huber loss for regression, logistic loss and squared hinge loss for classification and Cox proportional hazard model (see Section 5.5).

- We show improvements of our proposed screening framework across various datasets, a wide range of (λ_0, λ_2) and different branching strategies (see Section 5.6). In particular, our screening procedure can lead to up to 3 times runtime improvements for easy problems; it can help explore much more nodes (up to 2 times) and reduce the optimality gap within the time limit for hard problems.

Additional proofs can be found in Section 5.A. Additional technical and experimental details can be found in Appendix 5.B and Appendix 5.C.

Related work The notion of safe screening was originally proposed by earlier literature [95, 216, 86] for the ℓ_1-penalized regression Lasso problem [205]. It provably removes some zero parameters in the optimal solutions before the problem solving or during the optimization, and this can help significantly speed up the convex optimization solver. While in a similar spirit, our work is focused more on reducing the size of the tree and improving the branching, instead of simply accelerating the convex subproblem solvers.

In addition to the recent paper [9] mentioned above, [70] and [105] extend [9] in different directions. [70] consider a logistic regression case, and [105] consider the big-M formulation for linear regression and make an attempt to bring the screening idea at each node. However, here we consider a broader class of loss functions as well as the hybrid perspective formulation (which include the big-M as a special case). In addition, we also investigate into the interactions between branching rules and

the screening rule, and propose an enhanced screening in the cases where the strong branching rules are deployed.

Beyond papers mentioned above, there is also a lot of work on global optimization approaches [28, 30, 31, 222, 67] as well as fast approximate solvers [184, 72, 8, 18, 36] for sparse regression. Moreover, there is a rich body of work on solving mixed integer nonlinear programs (MINLPs) using BnB [146, 22] or outer approximations [80]. For more details, one can refer to [120] and the references therein. In this chapter, we focus on developing a safe screening rule within the BnB solver to improve the global optimization solver for Problem (5.1).

Notations For a vector $\boldsymbol{\beta} \in \mathbb{R}^p$, we use $\|\boldsymbol{\beta}\|_0$ to denote the number of nonzeros in $\boldsymbol{\beta}$; $\|\boldsymbol{\beta}\|$ to denote the Euclidean norm of $\boldsymbol{\beta}$. We will use $\dagger$ and $*$ as superscripts to denote optimal solutions/values to mixed integer problems and convex problems, respectively.

5.2 Problem formulations and branch-and-bound solver

Perspective formulation In this chapter, we adopt the hybrid perspective reformulation proposed by [120], given by

$$
\begin{aligned}
F_I^\dagger := \min_{\boldsymbol{\beta},\boldsymbol{z}} \;\; & F(\boldsymbol{\beta},\boldsymbol{z}) := f(\boldsymbol{\beta}) + \sum_{i=1}^{p} \big(\lambda_0 z_i + \lambda_2 \beta_i^2 / z_i\big) \qquad (P_I) \\
\text{s.t.} \;\; & (\boldsymbol{\beta},\boldsymbol{z}) \in \mathcal{C}_M := \{(\boldsymbol{\beta},\boldsymbol{z}) : |\beta_i| \le M z_i, \; \forall i \in [p]\} \\
& \boldsymbol{z} \in \{0,1\}^p.
\end{aligned}
$$

for some $M > 0$. Here, we adopt the convention that $0/0 = 0$, $0 \cdot \infty = 0$, and $x/0 = \infty$ for $x > 0$. We denote by $(\boldsymbol{\beta}^\dagger, \boldsymbol{z}^\dagger)$ an optimal solution to (P_I).

[120] show that compared to perspective reformulation [88, 4, 101, 72] and big-M reformulation [28, 222, 161, 116] used before, the hybrid version leads to tighter

reformulation in some circumstances. Note that when $M = \infty$ and $\lambda_2 > 0$, (P_I) reduces to perspective formulation; when $\lambda_2 = 0$ and $M < \infty$, this becomes the big-M formulation. In the setting of screening, [9, 70] use the perspective formulation, and [105] use the big-M formulation. Our proposed framework works on the hybrid version, which can also handle these cases.

Relaxation formulation The branch-and-bound solver relies heavily on solving the convex (interval) relaxation problem as follows

$$F^*(\mathcal{Z}) := \min_{\boldsymbol{\beta}, \boldsymbol{z}} \; F(\boldsymbol{\beta}, \boldsymbol{z}), \quad \text{s.t.} \;\; (\boldsymbol{\beta}, \boldsymbol{z}) \in \mathcal{C}_M, \;\; \boldsymbol{z} \in \mathcal{Z}, \tag{$P_{\mathcal{Z}}$}$$

where $\mathcal{Z} = \prod_{i=1}^{p} [\underline{z}_i, \bar{z}_i]$ for some $\underline{z}_i, \bar{z}_i \in \{0, 1\}^p$. Denote by $(\boldsymbol{\beta}^*(\mathcal{Z}), \boldsymbol{z}^*(\mathcal{Z}))$ an optimal solution to $(P_{\mathcal{Z}})$.

Branch-and-Bound solver In Algorithm 5.1, we provide an overview of the BnB solver with the approximate subproblem solver and screening tests. The high-level ideas of the algorithm is as follows: The algorithm starts with solving the the interval relaxation at the root node r, where all binary variables z_i's are relaxed to the interval $[0, 1]$, i.e. $(P_{\mathcal{Z}})$ with $\mathcal{Z} = \mathcal{Z}_r := [0, 1]^p$. Then, the algorithm selects a branching variable z_j, and creates two child nodes (new convex optimization problems), one with $\mathcal{Z} = \mathcal{Z}_r \cap \{z_j = 0\}$ and the other with $\mathcal{Z} = \mathcal{Z}_r \cap \{z_j = 1\}$. The algorithm grows the binary trees recursively until either of two conditon are met: (i) the optimal solution to the current node relaxation problem is integral (Line 6 in Algorithm 5.1); or (ii) the objective of the current optimization problem exceeds the best available upper bound $\bar{F}_I$ on (P_I) (Line 8). The algorithm terminates if the relative optimality gap, defined as $(\bar{F}_I - \underline{F}_I)/\underline{F}_I$ is smaller than a pre-specified tolerance.

Our proposals Following [120], for scalability considerations, we apply the primal first-order methods, such as proximal gradient descent (PGD) and coordinate descent (CD) to solve the relaxation subproblems. Dual bounds for the node relaxation problems are required to prune nodes as well as perform screening and branching

Algorithm 5.1 An overview of proposed BnB solver

Input: Set $\bar{F}_I = \infty$, $\underline{F}_I = -\infty$ and initialize the set of nodes to be solved $\mathcal{H} = \emptyset$
Output: An integral solution

1: Add the root node (P_r) to the set $\mathcal{H} = \mathcal{H} \cup \{P_r\}$
2: **while** $\mathcal{H} \neq \emptyset$ **do**
3: Remove a problem $(P_{\mathcal{Z}})$ from the set: $\mathcal{H} = \mathcal{H} - \{P_{\mathcal{Z}}\}$
4: Solve $(P_{\mathcal{Z}})$ inexactly and let $(\hat{\boldsymbol{\beta}}(\mathcal{Z}), \hat{\boldsymbol{z}}(\mathcal{Z}))$ be an approximate solution
5: Update the incumbent solution if a better integral solution is found, **continue**
6: Compute a lower bound $\underline{F}(\mathcal{Z})$ on $F^*(\mathcal{Z})$, based on $(\hat{\boldsymbol{\beta}}(\mathcal{Z}), \hat{\boldsymbol{z}}(\mathcal{Z}))$
7: Update lower bound $\underline{F}_I$ on MIP, based on $F(\mathcal{Z})$ under some condition
8: **If** $\underline{F}(\mathcal{Z}) > \bar{F}_I$, the current node can be pruned; **otherwise**, perform screening and branching procedure, and add new nodes to $\mathcal{H}$ if any
9: **end while**

procedure. We develop an efficient method for obtaining dual bounds from the primal solutions, which will be detailed in Section 5.3. Based on these lower bounds, we develop a novel screening procedure in combination with the branching procedure in BnB, and this will be discussed in Section 5.4.

5.3 Characterizations of relaxation subproblems

In this section, we provide characterizations of relaxation subproblems $(P_{\mathcal{Z}})$ for both root relaxation and the relaxations at general nodes. To simplify the presentation, we will focus mainly on the root relaxation problem, and present its properties including the optimality conditions (in Section 5.3.1), primal-dual relationship (in Section 5.3.2) and dual bounds (in Section 5.3.3). We then present the properties of the relaxation problems at general nodes in Section 5.3.4.

To this end, we first introduce some notations in this section. Denote by (P_r) the root relaxation problem, i.e. the problem $(P_{\mathcal{Z}})$ with $\mathcal{Z} = \mathcal{Z}_r = [0,1]^p$, which can be written as

$$F^*(\mathcal{Z}) := \min_{\boldsymbol{\beta}, \boldsymbol{z}} \; F(\boldsymbol{\beta}, \boldsymbol{z}), \quad \text{s.t.} \;\; (\boldsymbol{\beta}, \boldsymbol{z}) \in \mathcal{C}_M, \;\; \boldsymbol{z} \in [0,1]^p. \tag{P_r}$$

In Section 5.3.1 to 5.3.3, we will focus on the root relaxation problem, and for brevity, we denote by $F_r^* = F^*(\mathcal{Z}_r)$, $(\boldsymbol{\beta}^*, \boldsymbol{z}^*) = (\boldsymbol{\beta}^*(\mathcal{Z}_r), \boldsymbol{z}^*(\mathcal{Z}_r))$.

5.3.1 Optimality conditions

We first derive the optimality conditions for the root relaxation problem (P_r). We consider the Lagrangian dual by making use of the following identity: for any $z_i \geq 0$,

$$\frac{\beta_i^2}{z_i} = \max_{u_i} \; u_i\beta_i - \frac{u_i^2}{4}z_i, \tag{5.2}$$

and dualizing the constraint $|\beta_i| \leq Mz_i$ defined in $\mathcal{C}_M$ at the same time. More specifically, the Lagrangian can be written as

$$L(\boldsymbol{\beta}, \boldsymbol{z}; \boldsymbol{u}, \boldsymbol{v}) := f(\boldsymbol{\beta}) + \sum_{i=1}^{p} \left(\lambda_0 z_i + \lambda_2 u_i\beta_i - \lambda_2 \frac{u_i^2}{4} z_i + v_i\beta_i - M|v_i|z_i \right), \tag{5.3}$$

and the domain of $\boldsymbol{z}$ is $\mathcal{Z}_r = [0,1]^p$.

Proposition 5.1. *The strong duality holds for the root relaxation problem (P_r):*

$$\min_{(\boldsymbol{\beta},\boldsymbol{z})\in\mathcal{C}_M, \boldsymbol{z}\in\mathcal{Z}_r} \; F(\boldsymbol{\beta}, \boldsymbol{z}) = \min_{\boldsymbol{\beta}\in\mathbb{R}^p, \boldsymbol{z}\in\mathcal{Z}_r} \max_{\boldsymbol{u},\boldsymbol{v}} \; L(\boldsymbol{\beta}, \boldsymbol{z}; \boldsymbol{u}, \boldsymbol{v}) = \max_{\boldsymbol{u},\boldsymbol{v}} \min_{\boldsymbol{\beta}\in\mathbb{R}^p, \boldsymbol{z}\in\mathcal{Z}_r} \; L(\boldsymbol{\beta}, \boldsymbol{z}; \boldsymbol{u}, \boldsymbol{w}). \tag{5.4}$$

Furthermore, $(\boldsymbol{\beta}^*, \boldsymbol{z}^*)$ and $(\boldsymbol{u}^*, \boldsymbol{v}^*)$ are an optimal primal-dual pair, if and only if the following KKT condition holds:

$$0 = \nabla_i f(\boldsymbol{\beta}^*) + \lambda_2 u_i^* + v_i^* \tag{5.5a}$$

$$z_i^* \in \begin{cases} \{1\} & \text{if } \lambda_0 - \frac{\lambda_2 (u_i^*)^2}{4} - M|v_i^*| < 0 \\ \{0\} & \text{if } \lambda_0 - \frac{\lambda_2 (u_i^*)^2}{4} - M|v_i^*| > 0 \\ [0,1] & \text{otherwise} \end{cases} \tag{5.5b}$$

$$0 = 2\beta_i^* - u_i^* z_i^* \tag{5.5c}$$

$$v_i^* \in \begin{cases} (-\infty, \infty) & \text{if } z_i^* = 0 \\ \{0\} & \text{if } z_i^* > 0 \text{ and } |\beta_i^*| < Mz_i^* \\ (-\infty, 0] & \text{if } z_i^* > 0 \text{ and } \beta_i^* = -Mz_i^* \\ [0, \infty) & \text{if } z_i^* > 0 \text{ and } \beta_i^* = Mz_i^* \end{cases} \tag{5.5d}$$

$$(\boldsymbol{\beta}^*, \boldsymbol{z}^*) \in \mathcal{C}_M, \;\; \boldsymbol{z}^* \in \mathcal{Z}_r. \tag{5.5e}$$

Note that (5.5a) and (5.5c) are the first-order conditions wrt $\boldsymbol{\beta}$ and $\boldsymbol{u}$, respectively; (5.5b) includes both first-order condition wrt $\boldsymbol{z}$ and complementary slackness wrt the constraint $\boldsymbol{z} \in [0,1]^p$; (5.5d) includes both first-order condition wrt v_i and complementary slackness wrt the constraint $|\beta_i| \leq M z_i$ and the dual v_i; (5.5e) is the primal feasibility condition.

The proof of strong duality is based on Fenchel Duality Theorem (e.g. [193, Corollary 31.2.1]). The details can be found in Section 5.A.1.

5.3.2 Optimal dual variables

From the KKT conditions (5.5), the following proposition provides an approach of computing a optimal dual solution $(\boldsymbol{u}^*, \boldsymbol{v}^*)$ based on an optimal primal solution $(\boldsymbol{\beta}^*, \boldsymbol{z}^*)$:

Proposition 5.2. *Given an optimal primal solution $(\boldsymbol{\beta}^*, \boldsymbol{z}^*)$ to (P_r), then $(\boldsymbol{u}^*, \boldsymbol{v}^*)$ given in* (5.6) *is an optimal dual solution corresponding to it.*

$$u_i^* = -\min\{|\nabla_i f(\boldsymbol{\beta}^*)|, 2\lambda_2 M\} \operatorname{sign}(\nabla_i f(\boldsymbol{\beta}^*))/\lambda_2 \tag{5.6a}$$

$$v_i^* = -(|\nabla_i f(\boldsymbol{\beta}^*)| - 2\lambda_2 M)_+ \operatorname{sign}(\nabla_i f(\boldsymbol{\beta}^*)). \tag{5.6b}$$

Let $\Delta_i^* = \lambda_0 - \frac{\lambda_2 (u_i^*)^2}{4} - M|v_i^*|$, then we have

$$\Delta_i^* = h(\nabla_i f(\boldsymbol{\beta}^*)), \tag{5.7}$$

where

$$h(\alpha) = \begin{cases} \lambda_0 - \frac{\alpha^2}{4\lambda_2} & \text{if } |\alpha| \leq 2\lambda_2 M \\ \lambda_0 + \lambda_2 M^2 - M|\alpha| & \text{otherwise} \end{cases}. \tag{5.8}$$

Proof. For each i, we consider two cases based on the sign of z_i^*: $z_i^* = 0$ and $z_i^* \neq 0$.

Case 1: $z_i^* = 0$. In this case, we know $\beta_i^* = 0$ from (5.5c), and v_i^* is free from (5.5d). In addition, by (5.5a) and (5.5b), there exists u_i^* and v_i^* such that $\lambda_2 u_i^* + v_i^* = -\nabla_i f(\boldsymbol{\beta}^*)$ and $\lambda_0 - \frac{\lambda_2 (u_i^*)^2}{4} - M|v_i^*| \geq 0$. Therefore, we can find one

feasible u_i^* and v_i^* by solving the following optimization problem

$$(u_i^*, v_i^*) = \arg\max_{u,v} \ \lambda_0 - \frac{\lambda_2 (u_i^*)^2}{4} - M|v_i^*|, \ \text{ s.t. } \ \lambda_2 u_i^* + v_i^* = -\nabla_i f(\boldsymbol{\beta}^*), \tag{5.9}$$

which leads to (5.6). By the maximality of (5.9), we know that $\lambda_0 - \frac{\lambda_2 (u_i^*)^2}{4} - M|v_i^*| \geq 0$, i.e. (5.5b) holds; other conditions hold by definition, so (β_i^*, z_i^*) and (u_i^*, v_i^*) satisfy (5.5).

Case 2: $z_i^* > 0$. In this case, from the KKT condition (5.5), we know (u_i^*, v_i^*) is uniquely determined by (5.5a) and (5.5c), i.e.

$$u_i^* = \frac{2\beta_i^*}{z_i^*}, \quad \text{and} \quad v_i^* = -\lambda_2 u_i^* - \nabla_i f(\boldsymbol{\beta}^*) = -2\lambda_2 \frac{\beta_i^*}{z_i^*} - \nabla_i f(\boldsymbol{\beta}^*). \tag{5.10}$$

Now it remains to show that (5.6) leads to (5.10) in this case.

If $|\beta_i^*| = M z_i$, by (5.10), we have $u_i^* = 2M \operatorname{sign}(\beta_i^*)$ and

$$v_i^* + 2\lambda_2 M \operatorname{sign}(\beta_i^*) = -\nabla_i f(\boldsymbol{\beta}^*). \tag{5.11}$$

It follows from (5.5d) that $v_i^* \beta_i^* \geq 0$. Therefore, by taking the signs of both sides of (5.11), we have $\operatorname{sign}(\beta_i^*) = -\operatorname{sign}(\nabla_i f(\boldsymbol{\beta}^*))$; by taking the absolute values of both sides, we have $|v_i^*| + 2\lambda_2 M = |\nabla_i f(\boldsymbol{\beta}^*)|$. This implies $|\nabla_i f(\boldsymbol{\beta}^*)| \geq 2\lambda_2 M$. By some calculations, it is not hard to see that (5.6) leads to (5.10).

If $|\beta_i^*| < M z_i$, then by (5.5d), $|u_i| < 2M$. It follows from (5.5d) that $v_i^* = 0$. Combining this with (5.10), we get $\nabla_i f(\boldsymbol{\beta}^*) = -\lambda_2 u_i$ and thus $|\nabla_i f(\boldsymbol{\beta}^*)| < 2\lambda_2 M$. By some calculations, it is not hard to see that (5.6) leads to (5.10).

Hence, we have shown that in both cases, (5.6) gives a dual variable $(\boldsymbol{u}^*, \boldsymbol{v}^*)$ that along with $(\boldsymbol{\beta}^*, \boldsymbol{z}^*)$ satisfies the KKT conditions (5.5).

Finally, (5.7) follows from direct calculations. □

5.3.3 Dual bounds

In this section, we derive a dual bound on the relaxation problem (P_r), i.e. a lower bound on F_r^*. This is used in the BnB solver for pruning the search space; see Line 8 in Algorithm 5.1.

Proposition 5.3. *Given any $(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{z}}) \in \mathbb{R}^p \times \mathcal{Z}_r$ (not necessarily feasible to (P_r)), let*

$$\hat{\Delta}_i = h(\nabla_i f(\hat{\boldsymbol{\beta}})) = \begin{cases} \lambda_0 - \frac{|\nabla_i f(\hat{\boldsymbol{\beta}})|^2}{4\lambda_2} & \text{if } |\nabla_i f(\hat{\boldsymbol{\beta}})| \le 2\lambda_2 M \\ \lambda_0 + \lambda_2 M^2 - M|\nabla_i f(\hat{\boldsymbol{\beta}})| & \text{otherwise} \end{cases}, \tag{5.12}$$

and define

$$\underline{F}_r = f(\hat{\boldsymbol{\beta}}) - \langle \nabla f(\hat{\boldsymbol{\beta}}), \hat{\boldsymbol{\beta}} \rangle - \sum_{j=1}^{p} (-\hat{\Delta}_j)_+. \tag{5.13}$$

Then, we have $F_r^* \ge \underline{F}_r$. Furthermore, when $(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{z}})$ is an optimal solution to (P_r), then $F_r^* = \underline{F}_r$.

Proof. Let $(\tilde{\boldsymbol{u}}, \tilde{\boldsymbol{v}})$ be any dual variable, and denote by $\tilde{\Delta}_i = \lambda_0 - \frac{\lambda_2 \tilde{u}_i^2}{4} - M|\tilde{v}_i|$.

We first show that for any $\tilde{\boldsymbol{\beta}}$ and $(\tilde{\boldsymbol{u}}, \tilde{\boldsymbol{v}})$ such that $\nabla f(\tilde{\boldsymbol{\beta}}) + \lambda_2 \tilde{\boldsymbol{u}} + \tilde{\boldsymbol{v}} = 0$, we have

$$\min_{\boldsymbol{\beta} \in \mathbb{R}^p, \boldsymbol{z} \in \mathcal{Z}_r} L(\boldsymbol{\beta}, \boldsymbol{z}; \tilde{\boldsymbol{u}}, \tilde{\boldsymbol{v}}) = f(\tilde{\boldsymbol{\beta}}) - \langle \nabla f(\tilde{\boldsymbol{\beta}}), \tilde{\boldsymbol{\beta}} \rangle - \sum_{j=1}^{p} (-\tilde{\Delta}_j)_+. \tag{5.14}$$

By the definition of the Lagrangian L, we can write it as

$$L(\boldsymbol{\beta}, \boldsymbol{z}; \tilde{\boldsymbol{u}}, \tilde{\boldsymbol{v}}) = f(\boldsymbol{\beta}) + \langle \lambda_2 \tilde{\boldsymbol{u}} + \tilde{\boldsymbol{v}}, \boldsymbol{\beta} \rangle + \sum_{j=1}^{p} \tilde{\Delta}_j z_j. \tag{5.15}$$

The minimization problem of L wrt $\boldsymbol{\beta} \in \mathbb{R}^p$ and $\boldsymbol{z} \in \mathcal{Z}_r$ is separable and can be decomposed into subproblems wrt $\boldsymbol{\beta}$ and z_j's.

For $z_j \in [0, 1]$, it is clear to see the subproblem has the minimizer $z_j^* = 0$ if $\tilde{\Delta}_j > 0$; $z_j^* = 1$ if $\tilde{\Delta}_j < 0$; $z_j^* \in (0, 1)$ if $\tilde{\Delta}_j = 0$. Thus, $\min_{z_j \in [0,1]} \tilde{\Delta}_j z_j = -(-\tilde{\Delta}_j)_+$.

As for $\boldsymbol{\beta}$, when $\nabla f(\tilde{\boldsymbol{\beta}}) + \lambda_2 \tilde{\boldsymbol{u}} + \tilde{\boldsymbol{v}} = 0$, i.e. $\nabla_{\boldsymbol{\beta}} L(\tilde{\boldsymbol{\beta}}, \boldsymbol{z}; \tilde{\boldsymbol{u}}, \tilde{\boldsymbol{v}}) = 0$, we have $\tilde{\boldsymbol{\beta}}$ is an optimal solution to the subproblem wrt $\boldsymbol{\beta}$. Thus, $\min_{\boldsymbol{\beta}} f(\boldsymbol{\beta}) + \langle \lambda_0 \tilde{\boldsymbol{u}} + \tilde{\boldsymbol{v}}, \boldsymbol{\beta} \rangle = f(\tilde{\boldsymbol{\beta}}) - \langle \nabla f(\tilde{\boldsymbol{\beta}}), \tilde{\boldsymbol{\beta}} \rangle$.

Therefore, combining these subproblems yields (5.14).

Hence, combining Proposition 5.1 with (5.14), we have

$$\begin{aligned} F_r^* &= \max_{\boldsymbol{u},\boldsymbol{v}} \min_{\boldsymbol{\beta}\in\mathbb{R}^p,\boldsymbol{z}\in\mathcal{Z}_r} L(\boldsymbol{\beta},\boldsymbol{z};\boldsymbol{u},\boldsymbol{v}) \\ &\geq \min_{\boldsymbol{\beta}\in\mathbb{R}^p,\boldsymbol{z}\in\mathcal{Z}_r} L(\boldsymbol{\beta},\boldsymbol{z};\tilde{\boldsymbol{u}},\tilde{\boldsymbol{v}}) \\ &= f(\tilde{\boldsymbol{\beta}}) - \langle \nabla f(\tilde{\boldsymbol{\beta}}), \tilde{\boldsymbol{\beta}}\rangle - \sum_{j=1}^{p} (-\tilde{\Delta}_j)_+ \end{aligned} \tag{5.16}$$

for any $\tilde{\boldsymbol{\beta}}$ and $(\tilde{\boldsymbol{u}},\tilde{\boldsymbol{v}})$ that satisfy $\nabla f(\tilde{\boldsymbol{\beta}}) + \lambda_2\tilde{\boldsymbol{u}} + \tilde{\boldsymbol{v}} = 0$.

Now it remains to choose suitable $\tilde{\boldsymbol{\beta}}$ and $(\tilde{\boldsymbol{u}},\tilde{\boldsymbol{v}})$ in (5.16), based on $(\hat{\boldsymbol{\beta}},\hat{\boldsymbol{z}})$.

Here, we simply take $\tilde{\boldsymbol{\beta}} = \hat{\boldsymbol{\beta}}$. As for $(\tilde{\boldsymbol{u}},\tilde{\boldsymbol{v}})$, we want to choose $(\tilde{\boldsymbol{u}},\tilde{\boldsymbol{v}})$ to make the lower bound given by (5.16) as large as possible, given $\tilde{\boldsymbol{\beta}} = \hat{\boldsymbol{\beta}}$ and $\nabla f(\hat{\boldsymbol{\beta}}) + \lambda_2\tilde{\boldsymbol{u}} + \tilde{\boldsymbol{v}} = 0$.

Denote by

$$\hat{\Delta}_j = \max_{\tilde{u}_j,\tilde{v}_j} \ \tilde{\Delta}_j = \lambda_0 - \frac{\lambda_2\tilde{u}_j^2}{4} - M|\tilde{v}_j|, \quad \text{s.t.} \ \ \nabla_j f(\hat{\boldsymbol{\beta}}) + \lambda_2\tilde{u}_j + \tilde{v}_j = 0, \tag{5.17}$$

and denote by $(\hat{u}_j,\hat{v}_j)$ the corresponding maximizer

$$\hat{u}_i = -\min\{|\nabla_i f(\hat{\boldsymbol{\beta}})|, 2\lambda_2 M\}\,\mathrm{sign}(\nabla_i f(\hat{\boldsymbol{\beta}}))/\lambda_2 \tag{5.18a}$$

$$\hat{v}_i = -(|\nabla_i f(\hat{\boldsymbol{\beta}})| - 2\lambda_2 M)_+\,\mathrm{sign}(\nabla_i f(\hat{\boldsymbol{\beta}})). \tag{5.18b}$$

The expression of $\hat{\Delta}_j$ is given by (5.12), and it follows from (5.16) that such choice of $(\hat{\boldsymbol{u}},\hat{\boldsymbol{v}})$ leads to the lower bound given in (5.13).

Finally, we show that $F_r^* = f(\boldsymbol{\beta}^*) - \langle \nabla f(\boldsymbol{\beta}^*), \boldsymbol{\beta}^*\rangle - \sum_{j=1}^{p}(-\Delta_j^*)_+$for any optimal solution to $(\boldsymbol{\beta}^*,\boldsymbol{z}^*)$. Notice that by the definition of $(\boldsymbol{u}^*,\boldsymbol{v}^*)$ in (5.6), we know that $\nabla f(\boldsymbol{\beta}^*) + \lambda_2\boldsymbol{u}^* + \boldsymbol{v}^* = 0$. Therefore, it follows from Proposition 5.1 and (5.14) that

$$F_r^* = \min_{\boldsymbol{\beta},\boldsymbol{z}} L(\boldsymbol{\beta},\boldsymbol{z};\boldsymbol{u}^*,\boldsymbol{v}^*) = f(\boldsymbol{\beta}^*) - \langle \nabla f(\boldsymbol{\beta}^*), \boldsymbol{\beta}^*\rangle - \sum_{j=1}^{p}(-\Delta_j^*)_+. \tag{5.19}$$

□

Remark 5.1. *Notice that the dual bound expression in* (5.13) *depends only on* $\hat{\boldsymbol{\beta}}$*, not on* $\hat{\boldsymbol{z}}$*. In fact, recall that*

$$F(\boldsymbol{\beta}, \boldsymbol{z}) := f(\boldsymbol{\beta}) + \sum_{i=1}^{p} \big(\lambda_0 z_i + \lambda_2 \beta_i^2 / z_i\big). \tag{5.20}$$

For any fixed β_i*, the minimization problem* (P_r) *wrt* z_i *has a solution*

$$z_i = \min\left\{1, \max\left\{|\beta_i|/M, |\beta_i|\sqrt{\lambda_2/\lambda_0}\right\}\right\}. \tag{5.21}$$

This holds in edge cases where $\lambda_2 = 0$ *or* $M = \infty$ *as well. Based on this relationship* (5.21)*, the relaxation problem can be reduced to an optimization problem wrt* $\boldsymbol{\beta}$ *on its own. Therefore, the dual bound* (5.13) *can also regarded as a lower bound on the reduced optimization problem, and this provides an explanation why the expression depends only on* $\hat{\boldsymbol{\beta}}$ *instead of* $(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{z}})$*.*

Additional details around this reduction can be found in [120], where they apply the first-order methods on the reduced problem to solve the relaxation problems. Here, we adopt similar approaches to get an approximate solution $\hat{\boldsymbol{\beta}}$*, and obtain the corresponding* $\hat{\boldsymbol{z}}$ *via* (5.21)*.*

Remark 5.2. *Denote by* $f^*(\boldsymbol{\alpha}) = \sup_{\boldsymbol{\beta}}\{\langle \boldsymbol{\alpha}, \boldsymbol{\beta}\rangle - f(\boldsymbol{\beta})\}$ *the Fenchel conjugate of* f*, then it can be shown that the dual of* (P_r) *is given by* $\max_{\boldsymbol{\alpha}}\ D(\boldsymbol{\alpha}) := -f^*(\boldsymbol{\alpha}) - \sum_{i=1}^{p}[-h(\alpha_i)]_+$*, and the dual bound* $\underline{F}_r$ *given in* (5.13) *is the value of the dual function* D *evaluated at* $\boldsymbol{\alpha} = \nabla f(\hat{\boldsymbol{\beta}})$*. Since this is not the focus of the work, we will not provide the proof for this claim.*

5.3.4 Node relaxations

In this section, we present the properties of the convex relaxation problem (P_v) at a general node v:

$$F^*(\mathcal{Z}) := \min_{\boldsymbol{\beta}, \boldsymbol{z}}\ F(\boldsymbol{\beta}, \boldsymbol{z}), \quad \text{s.t. } (\boldsymbol{\beta}, \boldsymbol{z}) \in \mathcal{C}_M,\ \boldsymbol{z} \in \mathcal{Z}_v. \tag{P_v}$$

Based on the definition of the branching process, the domain $\mathcal{Z}_v$ can be characterized as a triple of index sets $(\mathcal{F}_0, \mathcal{F}_1, \mathcal{R})$, where $\mathcal{F}_b$ denotes the indices of z_j's that are fixed to b for $b \in \{0,1\}$, and $\mathcal{R}$ denotes the indices of z_j's that are still relaxed to the interval $[0,1]$. With this notation, the Lagrangian of (P_v) can be written as

$$L_v(\boldsymbol{\beta}, \boldsymbol{z}; \boldsymbol{u}, \boldsymbol{v}) = f(\boldsymbol{\beta}) + \sum_{i \in \mathcal{F}_1} \left(\lambda_0 + \lambda_2 u_i \beta_i - \lambda_2 \frac{u_i^2}{4} - v_i \beta_i - M|v_i| \right) + \sum_{i \in \mathcal{R}} \left(\lambda_0 z_i + \lambda_2 u_i \beta_i - \lambda_2 \frac{u_i^2}{4} z_i - v_i \beta_i - M|v_i| z_i \right). \quad (5.22)$$

The strong duality holds for the root relaxation problem (P_r), i.e.

$$\min_{(\boldsymbol{\beta}, \boldsymbol{z}) \in \mathcal{C}_M, \boldsymbol{z} \in \mathcal{Z}_v} F(\boldsymbol{\beta}, \boldsymbol{z}) = \min_{\boldsymbol{\beta} \in \mathbb{R}^p, \boldsymbol{z} \in \mathcal{Z}_v} \max_{\boldsymbol{u}, \boldsymbol{v}} L(\boldsymbol{\beta}, \boldsymbol{z}; \boldsymbol{u}, \boldsymbol{v}) = \max_{\boldsymbol{u}, \boldsymbol{v}} \min_{\boldsymbol{\beta} \in \mathbb{R}^p, \boldsymbol{z} \in \mathcal{Z}_v} L(\boldsymbol{\beta}, \boldsymbol{z}; \boldsymbol{u}, \boldsymbol{w}). \quad (5.23)$$

Furthermore, $(\boldsymbol{\beta}^*, \boldsymbol{z}^*)$ and $(\boldsymbol{u}^*, \boldsymbol{v}^*)$ are an optimal primal-dual pair, if and only if the following KKT condition holds:

$$0 = \nabla_i f(\boldsymbol{\beta}^*) + \lambda_2 u_i^* + v_i^* \quad (5.24a)$$

$$z_i^* \in \begin{cases} \{1\} & \text{if } (i \in \mathcal{F}_1) \text{ or } (i \in \mathcal{R} \text{ and } \lambda_0 - \frac{\lambda_2 (u_i^*)^2}{4} - M|v_i^*| < 0) \\ \{0\} & \text{if } (i \in \mathcal{F}_0) \text{ or } (i \in \mathcal{R} \text{ and } \lambda_0 - \frac{\lambda_2 (u_i^*)^2}{4} - M|v_i^*| > 0) \\ [0,1] & \text{otherwise} \end{cases} \quad (5.24b)$$

$$0 = 2\beta_i^* - u_i^* z_i^* \quad (5.24c)$$

$$v_i^* \in \begin{cases} (-\infty, \infty) & \text{if } z_i^* = 0 \\ \{0\} & \text{if } z_i^* > 0 \text{ and } |\beta_i^*| < M z_i^* \\ (-\infty, 0] & \text{if } z_i^* > 0 \text{ and } \beta_i^* = -M z_i^* \\ [0, \infty) & \text{if } z_i^* > 0 \text{ and } \beta_i^* = M z_i^* \end{cases} \quad (5.24d)$$

$$(\boldsymbol{\beta}^*, \boldsymbol{z}^*) \in \mathcal{C}_M, \ \boldsymbol{z}^* \in \mathcal{Z}_v. \quad (5.24e)$$

Proposition 5.4. *Given any $(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{z}}) \in \mathbb{R}^p \times [0,1]^p$ (not necessarily feasible to (P_v)),*

for any $i \notin \mathcal{F}_0$, define

$$\hat{\Delta}_i = h(\nabla_i f(\hat{\boldsymbol{\beta}})) = \begin{cases} \lambda_0 - \frac{|\nabla_i f(\hat{\boldsymbol{\beta}})|^2}{4\lambda_2} & \text{if } |\nabla_i f(\hat{\boldsymbol{\beta}})| \leq 2\lambda_2 M \\ \lambda_0 + \lambda_2 M^2 - M|\nabla_i f(\hat{\boldsymbol{\beta}})| & \text{otherwise} \end{cases}, \quad (5.25)$$

and define

$$\underline{F}_v = f(\hat{\boldsymbol{\beta}}) - \langle \nabla f(\hat{\boldsymbol{\beta}}), \hat{\boldsymbol{\beta}} \rangle - \sum_{j \in \mathcal{R}} (-\hat{\Delta}_j)_+ + \sum_{j \in \mathcal{F}_1} \hat{\Delta}_j. \quad (5.26)$$

Then, we have $F_v^* \geq \underline{F}_v$. Furthermore, when $(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{z}})$ is an optimal solution to (P_v), then $F_v^* = \underline{F}_v$.

The proof of strong duality, KKT conditions and dual bounds for node relaxation is similar to that for root relaxation, and a proof sketch is provided in Section 5.A.1.

5.4 Screening framework for the BnB solver

The screening procedure is based on the key idea presented in the following lemma:

Lemma 5.1. *For any convex set $\mathcal{Z} \subseteq [0,1]^p$, let $\underline{F}(\mathcal{Z})$ be a lower bound on $F^*(\mathcal{Z})$, and $\bar{F}_I$ an upper bound on $F_I^\dagger$. Then, $\underline{F}(\mathcal{Z}) > \bar{F}_I$ implies $\boldsymbol{z}^\dagger \notin \mathcal{Z}$ for any optimal solution $(\boldsymbol{\beta}^\dagger, \boldsymbol{z}^\dagger)$ to (P_I).*

Proof. By definitions of $\underline{F}(\mathcal{Z})$ and $\bar{F}_I$, we have $F^*(\mathcal{Z}) \geq \underline{F}(\mathcal{Z}) > \bar{F}_I \geq F_I^\dagger$. By definition of $(\boldsymbol{\beta}^\dagger, \boldsymbol{z}^\dagger)$, we know that $F_I^\dagger = F(\boldsymbol{\beta}^\dagger, \boldsymbol{z}^\dagger)$. If $\boldsymbol{z}^\dagger \in \mathcal{Z}$, since $F^*(\mathcal{Z})$ is the optimal value of F over $\boldsymbol{z} \in \mathcal{Z}$, then we would have $F_I^\dagger = F(\boldsymbol{\beta}^\dagger, \boldsymbol{z}^\dagger) \geq F^*(\mathcal{Z})$, which yield contradiction. Therefore, we must have $\boldsymbol{z}^\dagger \notin \mathcal{Z}$. □

5.4.1 Screening at the root

We discuss how to perform safe screening rule at the root node, i.e. identifying the indices j's of which $z_j^\dagger = 0$ or $z_j^\dagger = 1$ for sure.

To this end, we first introduce some notations for this section. Recall that $\mathcal{Z}_r = [0,1]^p$ is the domain of $\boldsymbol{z}$ at the root relaxation, and $F_r^* = F(\mathcal{Z})$ is the optimal value

of the root relaxation. For any $j \in [p]$ and $b \in \{0,1\}$, we denote by $F_r^*(z_j = b) = F^*(\mathcal{Z}_r \cap \{z_j = b\})$, and we use $\underline{F}_r(z_j = b)$ to denote a lower bound on $F_r^*(z_j = b)$.

Corollary 5.1. *For any $j \in [p]$ and any $b \in \{0,1\}$, let $\underline{F}_r(z_j = 1-b)$ be a lower bound on $F_r^*(z_j = 1-b)$. If $\underline{F}_r(z_j = 1-b) > \bar{F}_I$, then we have $z_j^\dagger = b$ for any optimal solution $(\boldsymbol{\beta}^\dagger, \boldsymbol{z}^\dagger)$ to (P_I).*

Proof. This follows directly from Lemma 5.1 by invoking $\mathcal{Z} = \mathcal{Z}_r \cap \{z_j = 1-b\}$. □

Proposition 5.5. *Given any $(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{z}}) \in \mathbb{R}^p \times [0,1]^p$, define $\hat{\Delta}_i$ and $\underline{F}_r$ as in (5.12) and (5.13). For any $j \in [p]$ and $b \in \{0,1\}$, we have*

$$F_r^*(z_j = 1-b) \geq \underline{F}_r + \max\{(1-2b)\hat{\Delta}_j, 0\} > \bar{F}_I \implies z_j^\dagger = b. \tag{5.27}$$

Proof. It follows from Proposition 5.3 that

$$\underline{F}_r = f(\boldsymbol{\beta}) - \langle \nabla f(\boldsymbol{\beta}), \boldsymbol{\beta} \rangle - \sum_{i=1}^{p} (-\hat{\Delta}_i)_+. \tag{5.28}$$

Now consider the problem $(P_{\mathcal{Z}})$ with $\mathcal{Z} = \mathcal{Z}_r \cap \{z_j = 1-b\}$ and it corresponds to $\mathcal{F}_{1-b} = \{j\}$, $\mathcal{F}_b = \emptyset$ and $\mathcal{R} = [p] - \{j\}$. According to Proposition 5.4, we have

$$F_r^*(z_j = 1) \geq f(\boldsymbol{\beta}) - \langle \nabla f(\boldsymbol{\beta}), \boldsymbol{\beta} \rangle - \sum_{i \neq j} (-\hat{\Delta}_i)_+ + \Delta_j = \underline{F}_r + (\hat{\Delta}_j)_+ \tag{5.29a}$$

$$F_r^*(z_j = 0) \geq f(\boldsymbol{\beta}) - \langle \nabla f(\boldsymbol{\beta}), \boldsymbol{\beta} \rangle - \sum_{i \neq j} (-\hat{\Delta}_i)_+ = \underline{F}_r + (-\hat{\Delta}_j)_+, \tag{5.29b}$$

i.e. for any $j \in [p]$ and $b \in \{0,1\}$,

$$F_r^*(z_j = 1-b) \geq \underline{F}_r + \max\{(1-2b)\hat{\Delta}_j, 0\}. \tag{5.30}$$

Combining this with Corollary 5.1, we complete the proof. □

The following proposition shows that the screening rules presented in [9, 70] are special cases of our rule presented in Proposition 5.5. Its proof can be found in Section 5.B.1.1.

Proposition 5.6. *Proposition 5.5 recovers the screening rule given in [9, 70], when we take $M = \infty$ and $(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{z}}) = (\boldsymbol{\beta}^*, \boldsymbol{z}^*)$ as an optimal solution to (P_r), in the cases of linear regression and logistic regression, respectively.*

Our screening rule extends [9, 70] in several aspects: (a) apart from squared-error loss and logistic loss, our screening rule works for a general differentiable loss function; (b) as for the MIP reformulation, we consider a hybrid version of big-M and perspective reformulations [120] that encompasses both reformulations as special cases; (c) our screening rule works with an approximate solution to the root relaxation problem as opposed to an optimal solution. This allows us to speed up the BnB algorithm by solving the relaxation problem inexactly while still being able to perform the screening tests. In addition, with approximate solutions $(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{z}})$ satisfying the relationship (5.21), the screening may also happen when $\hat{z}_j \in (0, 1)$, while the screening never happens when $z_j^* \in (0, 1)$ if we take the optimal solution $(\boldsymbol{\beta}^*, \boldsymbol{z}^*)$.

To illustrate this, let's take a closer look into the case where we take $(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{z}}) = (\boldsymbol{\beta}^*, \boldsymbol{z}^*)$ as an optimal solution to (P_r). Let Δ_j^* be defined as in (5.7), and F_r^* be the optimal value to (P_r). Since any feasible solution to (P_I) is also feasible to (P_r), we have $F_r^* \leq \bar{F}_I$. Then, for any $j \in [p]$ and $b \in \{0, 1\}$ we have

$$F_r^* + \max\{(1-2b)\Delta_j^*, 0\} > \bar{F}_I \quad \overset{\text{(i)}}{\iff} \quad F_r^* + (1-2b)\Delta_j^* > \bar{F}_I \tag{5.31a}$$

$$\begin{cases} \overset{\text{(ii)}}{\implies} (1-2b)\Delta_j^* > 0 \overset{\text{(iii)}}{\implies} z_j^* = b \\ \overset{\text{(iv)}}{\implies} z_j^\dagger = b \end{cases} \tag{5.31b}$$

where (i) and (ii) is due to the fact that $F_r^* \leq \bar{F}_I$; (iii) is by the complementary slackness condition (5.5b) in Proposition 5.1; (iv) is by the screening rule in Proposition 5.5.

On the other hand, if $z_j^* \in (0, 1)$, then we know $\Delta_j^* = 0$ by (5.5b), and since $F_r^* \leq \bar{F}_I$, no screening happens in this case.

Hence, combining above arguments, we know that when given an optimal solution to the root relaxation, the screening happens only when z_j^* is binary, and when it happens, we must have $z_j^\dagger = z_j^*$.

On the other hand, given an approximate solution $(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{z}})$ satisfying (5.21), screening might happen when $\hat{z}_j \in (0,1)$, because $\hat{\Delta}_j$ may not necessarily be 0 in this case. In Section 5.B.1.2, we provide examples of screening when $\hat{z}_j$ is non-binary.

5.4.2 Screening at a general node

In this section, we discuss how to perform screening rule at each node in the BnB tree. We first introduce some notations in this section.

Recall that for each node v in the tree, $\mathcal{Z}_v$ denotes the domain of $\boldsymbol{z}$ at this node, and $F_v^* = F(\mathcal{Z}_v)$ is the optimal value of the node relaxation problem. Recall that the domain $\mathcal{Z}_v$ can be also represented by the index set triple $(\mathcal{F}_0, \mathcal{F}_1, \mathcal{R})$. Now for any $j \in [p]$ and $b \in \{0,1\}$, we denote by $F_v^*(z_j = b) = F^*(\mathcal{Z}_v \cap \{z_j = b\})$, and we use $\underline{F}_v(z_j = b)$ to denote a lower bound on $F_v^*(z_j = b)$.

The following corollary, which is a corollary of Lemma 5.1, presents the key idea of screening at node in a BnB tree.

Corollary 5.2. *For any $j \in [p]$ and $b \in \{0,1\}$, let $\underline{F}_v(z_j = b)$ be a lower bound on $F_v^*(z_j = b)$. If $\underline{F}_v(z_j = b) > \bar{F}_I$, then we can safely fix $z_j = 1 - b$ under the node v in the BnB tree. If $\min\{\underline{F}_v(z_j = 0), \underline{F}_v(z_j = 1)\} > \bar{F}_I$, then we can prune the node v in the BnB tree.*

Proof. When $\underline{F}_v(z_j = b) > \bar{F}_I$, by Lemma 5.1, we know that $\boldsymbol{z}^\dagger \notin \mathcal{Z}_v \cap \{z_j = b\}$. Therefore, under the node v, we need only consider $\mathcal{Z}_v \cap \{z_j = 1 - b\}$ for the search space, i.e. we can safely fix $z_j = 1 - b$ under the node in the BnB tree.

When $\min\{\underline{F}_v(z_j = 0), \underline{F}_v(z_j = 1)\} > \bar{F}_I$, then it follows from Lemma 5.1 that $\boldsymbol{z}^\dagger \notin \mathcal{Z}_v \cap \{z_j = 0\}$ and $\boldsymbol{z}^\dagger \notin \mathcal{Z}_v \cap \{z_j = 1\}$, i.e. $\boldsymbol{z}^\dagger \notin \mathcal{Z}_v$. Hence, we can prune the node v in the BnB tree. □

Proposition 5.7. *Given any $(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{z}}) \in \mathbb{R}^p \times [0,1]^p$, define $\underline{F}_v$ and $\hat{\Delta}_i$ as in (5.25) and (5.26). For any $j \in \mathcal{R}$ and $b \in \{0,1\}$, we have*

$$F_v^*(z_j = 1-b) \geq \underline{F}_v + \max\{(1-2b)\hat{\Delta}_j, 0\} > \bar{F}_I \implies \textit{fix } z_j = b \textit{ under node } v. \quad (5.32)$$

Proof. Using similar arguments in the proof for Proposition 5.5, we have for any $j \in \mathcal{R}$ and $b \in \{0, 1\}$

$$F_v^*(z_j = 1 - b) \geq \underline{F}_v + \max\{(1 - 2b)\hat{\Delta}_j, 0\}. \tag{5.33}$$

Combining this with Corollary 5.2, we complete the proof. □

Remark 5.3 (A note on pruning)**.** *Compared to Corollary 5.2, Corollary 5.1 does not have the statement on the pruning, because* $\min\{\underline{F}_r(z_j = 0), \underline{F}_r(z_j = 1)\} > \bar{F}_I$ *never happens. In fact, for the feasible solution* $(\bar{\boldsymbol{\beta}}, \bar{\boldsymbol{z}})$ *that achieves the upper bound* $\bar{F}_I$*, either* $\bar{\boldsymbol{z}} \in \mathcal{Z}_r \cap \{z_j = 0\}$ *or* $\bar{\boldsymbol{z}} \in \mathcal{Z}_r \cap \{z_j = 1\}$ *holds; therefore,* $\bar{F}_I = F(\bar{\boldsymbol{\beta}}, \bar{\boldsymbol{z}}) \geq \min\{F_r^*(z_j = 0), F_r^*(z_j = 1)\} \geq \min\{\underline{F}_r(z_j = 0), \underline{F}_r(z_j = 1)\}$.

For a non-root node v*,* $\min\{\underline{F}_v(z_j = 0), \underline{F}_v(z_j = 1)\} > \bar{F}_I$ *can happen because the incumbent solution* $\bar{\boldsymbol{z}}$ *does not necessarily belong to* $\mathcal{Z}_v$*. That being said, if the node* v *is not pruned in Line 8 in Algorithm 5.1 before entering the screening procedure (i.e.* $\underline{F}_v \leq \bar{F}_I$*), then the screening rules in Proposition 5.7 cannot prune the node either, because at least one of* $(\hat{\Delta}_j)_+$ *and* $(-\hat{\Delta}_j)_+$ *is 0.*

5.4.3 Screening and branching procedures

In this section, we describe how the theory presented in Section 5.4.1 and Section 5.4.2 can be applied in the BnB framework in combination with the branching procedures and the advantages of such screening.

Algorithm 5.2 Screening and branching procedure

Input: Node v with domain $\mathcal{Z}_v$, and the index triple $\{\mathcal{F}_0^v, \mathcal{F}_1^v, \mathcal{R}^v\}$, an approximate solution $(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{z}})$ to the node relaxation (P_v)

1: Obtain $\hat{\Delta}_i$ based on (5.17) for any $i \in \mathcal{R}^v$

2: For $b \in \{0,1\}$, set $\hat{\mathcal{F}}_b^v = \mathcal{F}_b^v \cup \{j \in \mathcal{R} : (5.32) \text{ holds}\}$, $\hat{\mathcal{R}}^v = [p] - \hat{\mathcal{F}}_0^v - \hat{\mathcal{F}}_1^v$

3: Set $\mathcal{J} = \{j \in \hat{\mathcal{R}} : \hat{z}_j \notin \{0,1\}\}$ as branching candidates

4: Select $j \in \mathcal{J}$ as the branching variable based on the branching rule; (for certain rules) perform enhanced screening and update $(\hat{\mathcal{F}}_0^v, \hat{\mathcal{F}}_1^v, \hat{\mathcal{R}}^v)$ during branching

5: Branch the node v into $\mathsf{left}(v)$ and $\mathsf{right}(v)$ with $\mathcal{F}_0^{\mathsf{left}(v)} = \hat{\mathcal{F}}_0^v \cup \{j\}, \mathcal{F}_0^{\mathsf{right}(v)} = \hat{\mathcal{F}}_0^v, \mathcal{F}_1^{\mathsf{left}(v)} = \hat{\mathcal{F}}_1^v, \mathcal{F}_1^{\mathsf{right}(v)} = \hat{\mathcal{F}}_1^v \cup \{j\}, \mathcal{R}^{\mathsf{left}(v)} = \mathcal{R}^{\mathsf{right}(v)} = \hat{\mathcal{R}}^v - \{j\}$.

To this end, we outline the screening and branching procedure in Algorithm 5.2. The procedure is as follows: at each node v, we first compute an approximate solution $(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{z}})$ to the node relaxation (P_v) using the first-order methods (similar to [120]). We then compute $\hat{\Delta}_j$ using (5.17) (Line 1) and based on these, we can safely fix some variables under the current node (Line 2), according to Corollary 5.2. Here, $(\hat{\mathcal{F}}_0^v, \hat{\mathcal{F}}_1^v, \hat{\mathcal{R}}^v)$ refers to the updated index triple after screening, which will be inherited by the children of the current node. After screening, we perform the branching procedure (Lines 3-5). Following the usual branching procedure in BnB (see, e.g. [22]), we first consider all fractional $\hat{z}_j$ from the remaining relaxed variables $\hat{\mathcal{R}}^v$ (instead of $\mathcal{R}^v$) as branching candidates, and then select the branching variable based on a certain branching rule.

5.4.3.1 Branching strategies

Many branching strategies for BnB have been studied in literature, e.g. maximum fractional branching, random branching, pseudo-cost branching and strong branching [24, 5, 151, 177, 1]. More details on branching for BnB can be found in [2, 38, 22, 168, 120]. In this chapter, we consider and study 4 different branching rules in combination with our screening procedure:

1. *Maximum fractional branching.* Maximum fraction rule selects the branching

variable as $j = \arg\max_{i \in \mathcal{J}} \min\{\hat{z}_i - \lfloor \hat{z}_i \rfloor, \lceil \hat{z}_i \rceil - \hat{z}_i\}$. This is also called *most infeasible* or *most fractional* rule [22, 168]. It is one of the most commonly-used and simplest rules. In practice, however, this might not be as effective as random branching [2].

2. *Maximum z branching.* Maximum z rule determines the branching variable with the largest $\hat{z}_j$: $j = \arg\max_{i \in \mathcal{J}} \hat{z}_i$. This is the branching rule used by [165, 105], which may work well when the solution is sparse. This is also similar to the *least fractional* rule, which is often outperformed by other methods [177].

3. *Strong branching (linear score).* Strong branching rule branches on the variable that leads to most impact on the objective function. Specifically, let δF_i^1 and δF_i^0 be estimates for the increase in the optimal values $F_v^*(z_i = 1) - F_v^*$ and $F_v^*(z_i = 0) - F_v^*$ after branching $\hat{z}_i$ to 1 and 0, respectively. For each candidate variable, a score s_i is computed based on δF_i^b's, and the branching variable is selected as $j = \arg\max_{i \in \mathcal{J}} s_i$. A common formula to compute s_i is

$$s_i = \mu \min(\delta F_i^0, \delta F_i^1) + (1 - \mu) \max(\delta F_i^0, \delta F_i^1) \tag{5.34}$$

for some constant $\mu \in [0, 1]$ typically taken close to 1. This strategy was first proposed by [5], and it is one of the most successful branching methods. It often leads to a small search tree, However, such strategy usually requires solving $2|\mathcal{J}|$ that many subproblems at each node to provide good estimates δF_i^b, so it induces a lot more computation costs compared to other approaches [2]. To address high computational cost, we use a fast approximate version of strong branching, in which we take a full pass of coordinate descent from $(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{z}})$ to obtain $(\hat{\boldsymbol{\beta}}^{i,b}, \hat{\boldsymbol{z}}^{i,b})$, and take $\delta F_i^b = F(\hat{\boldsymbol{\beta}}^{i,b}, \hat{\boldsymbol{z}}^{i,b}) - F(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{z}})$ as the estimates. Due to the warm start, in practice, this approximation often leads to similar BnB trees compared to exact strong branching [120].

4. *Strong branching (product score).* Instead of using a linear score function (5.34),

[1] recommend using the following product score function for strong branching:

$$s_i = \max(\delta F_i^0, \epsilon) \cdot \max(\delta F_i^1, \epsilon), \tag{5.35}$$

for a small positive value ϵ. In a recent work, [69] has shown empirically strong branching with such product score dominates the one with linear score by a small margin for certain problems.

5.4.3.2 Enhanced screening for strong branching

For strong branching strategies, during the course of branching, we perform another round of screening which we call *enhanced screening*, update the index triple again and possibly pruning the node. Such extra round of screening is possible mainly due to the updated dual information provided by the subproblem solving during the strong branching process. Recall that to provide estimates for δF_i^b in strong branching, we need to (approximately) solve the convex relaxation problem $(P_{\mathcal{Z}})$ with $\mathcal{Z} = \mathcal{Z} \cap \{z_i = b\}$ for any $i \in \mathcal{J}$ and $b \in \{0, 1\}$. We can then utilize the tailored primal solution $(\hat{\boldsymbol{\beta}}^{i,b}, \hat{\boldsymbol{z}}^{i,b})$ to compute an updated (and highly likely improved) dual bound $\underline{F}_v(z_i = b)$ according to Proposition 5.4. Thus, by Corollary 5.2, we can perform another round of screening and possibly pruning based on the new lower bounds. Note that different from the circumstance described in Remark 5.3, the pruning is possible at the enhanced screening time because the dual bounds are computed based on different solutions instead of the same solution $(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{z}})$.

5.4.3.3 Effects of screening

As we conclude this section, we summarize the advantages of using screening as a procedure within BnB framework. As we can see in Algorithm 5.2, without screening procedure in Line 2, only one variable is removed from $\mathcal{R}$ to either $\mathcal{F}_0$ or $\mathcal{F}_1$ at each time of branching; while on the contrary, there can be a bunch of variables being fixed to 0 or 1 at each node. As a result, it will reduce effective dimensions of convex subproblems to be solved, and thus subproblem solving will be more efficient. It

might also reduce sizes of search trees, because $\mathcal{R}$ may converge faster to empty set and a smaller $\mathcal{R}$ may lead to improved lower bounds and thus earlier termination. Furthermore, the screening procedure can reduce the size of branching candidates (in Line 3) and improve the efficiency of branching, especially for strong branching rules, which requires solving as many subproblems as twice the number of candidates. Finally, enhanced screening can make use of the updated information given by strong branching rules, which can further improve the efficiency of BnB framework.

5.5 Applications

In this section, we provide some examples of potential applications of our branch-and-bound framework for the sparse learning problem. In what follows, we will introduce various applications in different statistical learning settings.

5.5.1 Regression

In this section, we introduce different objective functions for sparse regression problems. Let $\boldsymbol{X} \in \mathbb{R}^{n\times p}$ and $\boldsymbol{y} \in \mathbb{R}^p$ denote the data matrix of independent variables and the observations of dependent variables, respectively. We consider the following three objectives with different losses:

- **Squared error loss**: The squared error loss is most commonly used in the statistical learning; it leads to the least squares problem. The squared error objective can be written as

$$f(\boldsymbol{\beta}) = \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|^2. \tag{5.36}$$

- **Huber loss**: The Huber loss is an important and useful loss function in robust statistics literature [124]. It is defined as

$$h_\delta(t) = \frac{t^2}{2\delta}\mathbf{1}\{|t| < \delta\} + (|t| - \frac{\delta}{2})\mathbf{1}\{|t| \geq \delta\} \tag{5.37}$$

for some positive δ. The corresponding optimization objective given data $\boldsymbol{X}$ and $\boldsymbol{y}$ is

$$f(\boldsymbol{\beta}) = \sum_{j=1}^{n} h_\delta(y_j - \boldsymbol{x}_j^\top \boldsymbol{\beta}). \tag{5.38}$$

We note that when $\delta \downarrow 0$, the optimization objective is equivalent to the mean absolution error loss. Therefore, it can be seen as a smooth version of median regression.

- **Smooth quantile loss**: Smooth quantile loss is a smooth version of quantile regression [7]. Given $\delta > 0$ and $q \in (0, 1)$, we define

$$h_{\delta,q}(t) = \begin{cases} q|t| - \frac{\delta q^2}{2} & \text{if } t \leq -q\delta \\ \frac{t^2}{2\delta} & \text{if } -q\delta < t < (1-q)\delta \\ (1-q)|t| - \frac{\delta(1-q)^2}{2} & \text{if } t \geq (1-q)\delta \end{cases}. \tag{5.39}$$

The corresponding smooth quantile regression is given by

$$f(\boldsymbol{\beta}) = \sum_{j=1}^{n} h_{\delta,q}(y_j - \boldsymbol{x}_j^\top \boldsymbol{\beta}). \tag{5.40}$$

When $\delta \downarrow 0$, this converges to the quantile regression problem.

5.5.2 Binary classification

In this section, we consider different objective functions for sparse binary classification problems. Let $\boldsymbol{X} \in \mathbb{R}^{n \times p}$ and $\boldsymbol{y} \in \{-1, 1\}^p$ denote the data matrix of independent variables and the observations of dependent variables, respectively. We consider the following three objective with different losses:

- **Logistic loss**: The logistic regression [113] is one of the most commonly used classifiers in machine learning. The objective is given by

$$f(\boldsymbol{\beta}) = \sum_{j=1}^{n} \log(1 + \exp(-y_j \boldsymbol{x}_j^\top \boldsymbol{\beta})). \tag{5.41}$$

- **Squared hinge loss**: Support Vector Machien (SVM [209]) is another important methodology for classification. The objective is based on *hinge loss*, which can be written as $\sum_{j=1}^{n}(1-y_j\boldsymbol{x}_j^\top\boldsymbol{\beta})_+$. However, the hinge loss is not a continuously differentiable function. We consider a smooth alternative with squared hinge loss [226]:

$$f(\boldsymbol{\beta})=\sum_{j=1}^{n}(1-y_j\boldsymbol{x}_j^\top\boldsymbol{\beta})_+^2. \tag{5.42}$$

- **Huberized hinge loss**: We could also consider another smooth sibling for SVM with Huberized hinge loss [217]:

$$f(\boldsymbol{\beta})=\sum_{j=1}^{n}h_\delta((1-y_j\boldsymbol{x}_j^\top\boldsymbol{\beta})_+). \tag{5.43}$$

5.5.3 Multi-class logistic model

For the multi-class classification problem, we assume there is m categories and thus the target variable $\mathsf{Y}\in\mathcal{Y}=[m]$. The logistic model assumes that given features $\mathsf{X}\in\mathbb{R}^p$, the probability of falling into category l is

$$\mathbb{P}(\mathsf{Y}=l|\mathsf{X})=\frac{\exp(\boldsymbol{\beta}_l^\top\mathsf{X})}{\sum_{l'=1}^{m}\exp(\boldsymbol{\beta}_{l'}^\top\mathsf{X})},\quad\forall l\in[m], \tag{5.44}$$

where $\boldsymbol{\beta}=[\boldsymbol{\beta}_1,\ldots,\boldsymbol{\beta}_m]\in\mathbb{R}^{p\times m}$ is the coefficient matrix to be learned. Given the data $(\boldsymbol{X},\boldsymbol{y})$, the logistic loss is given by

$$f(\boldsymbol{\beta})=\sum_{l=1}^{m}\sum_{j:y_j=l}-\boldsymbol{\beta}_k^\top\boldsymbol{x}_j+\sum_{j=1}^{n}\log\left(\sum_{l=1}^{m}\exp(\boldsymbol{\beta}_l^\top\boldsymbol{x}_j)\right) \tag{5.45}$$

5.5.4 Cox's proportional hazards

The Cox proportional hazards model [57] is a survival model in statistics; see [204] for more details. Given observations $\{(\boldsymbol{x}_j,O_j,T_j)\}_{j\in[n]}\subseteq\mathbb{R}^p\times\{0,1\}\times\mathbb{R}_+$, the log-partial

likelihood of the model can be written as

$$f(\boldsymbol{\beta}) = \sum_{j:O_j=1} \left[-\boldsymbol{\beta}^\top \boldsymbol{x}_j + \log \left(\sum_{l:T_l \geq T_j} \exp(\boldsymbol{\beta}^\top \boldsymbol{x}_l) \right) \right]. \tag{5.46}$$

This is a smooth objective, which can be handled in our framework.

5.6 Numerical Experiments

We present numerical experiments to investigate the impact of different levels of deployment of our proposed screening rules on the efficiency of the BnB solver. We conduct experiments on synthetic and real-world datasets for both least square regression problems and logistic regression problems, whose formulations are presented in (5.36) and (5.41), respectively.

5.6.1 Experimental setup

All computations were carried out on the MIT Supercloud Cluster [190] on an Intel Xeon Platinum 8260 machine, with 2 CPUs and 8GB of RAM. Our algorithms were written in Python with critical code sections optimized using Numba [144]. Code for our experiments is available from the github repository `l0bnb-screen` available at: `https://github.com/wenyuC94/l0bnb-screen`.

Algorithms We conduct all experiments using the Branch-and-Bound solver designed in Algorithm 5.1, but with various branching strategies and different levels of screening. Specifically, as mentioned in Section 5.4.3, we consider four branching strategies: maximum fractional branching (MaxFrac), maximum z branching (MaxZ), and strong branching rules with linear score (Strong-L) and product score (Strong-P). Different levels of screening include (i) plain BnB solver with no screening (Plain); (ii) BnB solver with screening at root only (Root); (iii) BnB solver with screening at each node (Node); (iv) BnB solver with enhanced screening at each node (Node-

E). Here, "Plain" can be seen as a counterpart of `L0BnB` framework [120][1], where first-order methods and warm starts are tailored to the nonlinear BnB for sparse learning; "Root" is a counterpart of [9, 70]; "Node" and "Node-E" are our proposals, and "Node-E" only applies to strong branching rules (Strong-L and Strong-P).

All algorithms are run with time limit of 4 hours for synthetic datasets and 8 hours for real datasets. Following [120], we terminate algorithms once the optimality gap for MIO (defined as $(\bar{F}_I - \underline{F}_I)/\underline{F}_I$) is smaller than 1%, where $\underline{F}_I$ is the lower bound on the MIP. Additional BnB solver settings can be found in Appendix 5.C.1.

Datasets For synthetic datasets, we generate datasets with $n = 1000$ samples and $p = 1000$ features. Following prior works [116, 120], we draw the data matrix $\boldsymbol{X} \in \mathbb{R}^{n\times p}$ from a multivariate Gaussian distribution with zero mean and a covariance matrix Σ with unit diagonal and constant correlation $\Sigma_{ij} = \rho > 0$. The underlying truth $\boldsymbol{\beta}^c \in \mathbb{R}^p$ is a sparse vector with $k = 10$ equispaced nonzero entries all set to 1. For classification model, following [67, 70], we generate each coordinate $y_j \in \{-1, 1\}$ independently by sampling from a Bernoulli distribution with success probability $\mathbb{P}(y_j = 1|\boldsymbol{x}_j) = (1 + \exp(-s\boldsymbol{x}_j^\top \boldsymbol{\beta}^c))^{-1}$, where s is a scale parameter. Specifically, smaller values of s increase the variance in the response, while $s \to \infty$ generates linearly separable data. In our experiments, we consider Scale $s \in \{1, 3, 5\}$, and for each s, we simulate 10 synthetic datasets. Additional details along with the details on the regression dataset generation can be found in Appendix 5.C.1.1.

Beyond synthetic datasets, we also consider four real-world datasets from the UCI Machine Learning Repository [74]. We preprocess the data by mean-centering and normalizing the response and columns of the data matrix. To facilitate exploration of a wider range of regularization parameters (λ_0, λ_2), we apply our implementation of `L0Learn` [116] with suitable regularization parameters and exclude all features not in the support of the obtained solution. The reduced datasets are summarized below

- Dexter: classification problem with $n = 300$ and $p = 457$
- Arcene: classification problem with $n = 100$ and $p = 482$

[1]Here, we have switched off the active set heuristics.

- REJA: classification problem with $n = 1996$ and $p = 477$
- Crime: regression problem with $n = 1999$ and $p = 94$

Additional details on real datasets can be found in Appendix 5.C.1.2.

Choices of λ_0, λ_2 and M To examine the effects of our proposed screening rules, we run experiments for a wide range of (λ_0, λ_2), with a fixed $M = 1$ throughout. The way to select the grid is similar to the methodologies adopted by the earlier works [116, 120]. In brief, we apply fast approximate solvers to solve the problems with a wide range of grid points, find the statistically "optimal" grid point(s), and then take 100 grid points around the optimal one(s). The details can be found in Appendix 5.C.1.3.

5.6.2 Numerical results

For each case with a certain choice of (λ_0, λ_2) and the dataset, we first categorize it as "easy" or "hard", based on whether the plain BnB can achieve 1% optimality gap within the time limit using any of four branching rules. We report the performance of different levels of screening and branching strategies for easy and hard cases separately due to different patterns presented in the results.

Easy cases Table 5.1 reports the average runtimes and the average size of the BnB search trees over all easy cases for synthetic classification datasets. Figure 5-1 provides a straightforward comparison on the distributions of different screening levels. We can see that on easy problems, Root provides negligible improvement in efficiency when compared to Plain, while on average Node can be up to 3 times faster than Plain, indicating its capacity to improve the overall efficiency of the solver. We attribute this to the fact that node screening greatly reduces the number of relaxed indices at each node, thereby decreasing the time required to solve the lower bound at each node.

Furthermore, we find that Node-E is even more effective than Node, as it is able to fix even more variables, leading to a reduction in the size of the BnB tree and further

reducing the overall running time. In terms of the size of the BnB tree, neither Root nor Node is effective in reducing the number of nodes explored. In contrast, enhanced node screening (Node-E) can reduce the size of the tree by up to 50%.

We note that the box plots show some outliers, which indicates that the performance of BnB solver can vary significantly depending on the dataset and the regularization parameters.

Results for real datasets presented in Table 5.2 are consistent with the observations above. Our further experiments on synthetic regression datasets reported in Appendix 5.C.2 provide qualitatively similar conclusions.

Table 5.1: Average runtime and tree sizes for **easy case** of logistic regression problems. Results are averaged over different choices of λ_0, λ_2, and random seeds of synthetic datasets. The numbers in the bracket below "scale" indicate the number of easy cases and total cases, respectively. Here, Time refers to runtime in seconds; Size refers to the number of nodes in the BnB tree.

Rules		**MaxFrac**			**MaxZ**			**Strong-L**				**Strong-P**			
		Plain	Root	Node	Plain	Root	Node	Plain	Root	Node	Node-E	Plain	Root	Node	Node-E
scale=1 (742/1000)	Time	2021	2024	718	1982	1979	728	2408	2393	1181	864	2400	2386	1144	862
	Size	2968	2972	2976	2940	2941	2950	2950	2964	2991	1668	2889	2893	2942	1627
scale=3 (883/1000)	Time	386	390	131	371	375	124	410	414	170	146	413	399	166	142
	Size	424	423	428	416	415	417	413	418	426	280	414	412	417	271
scale=5 (908/1000)	Time	336	348	112	319	314	104	351	348	137	133	340	339	138	129
	Size	344	347	348	340	340	340	346	348	349	253	339	337	339	245

Table 5.2: Average runtime and tree sizes for **easy case** of real-world problems. Results are averaged over different choices of λ_0, λ_2. Details are given in the caption of Table 5.1.

Rules		**MaxFrac**			**MaxZ**			**Strong-L**				**Strong-P**			
		Plain	Root	Node	Plain	Root	Node	Plain	Root	Node	Node-E	Plain	Root	Node	Node-E
REJA (51/100)	Time	2925	2709	1021	3312	2936	1262	2502	2602	1096	344	2781	2551	1031	364
	Size	1225	1225	1127	1270	1273	1176	1052	1050	973	421	1040	1048	969	413
Dexter (55/100)	Time	4313	4333	1395	5055	4974	1846	4137	4267	1671	662	4123	4251	1506	624
	Size	3728	3782	3358	3811	3313	3467	2786	3022	2635	1034	2961	2961	2562	976
Arcene (54/100)	Time	3139	3216	1021	3686	3914	1368	3085	3043	1088	467	3101	3220	1274	433
	Size	2060	2066	1857	2119	2132	1931	1673	1674	1509	620	1675	1676	1497	614
Crime (97/100)	Time	3252	3137	2469	3195	3149	2582	2201	2209	1711	1574	2238	2211	1700	1656
	Size	12396	12236	11356	12568	12455	11360	8085	8149	6802	6628	8226	8223	6883	6667

Hard cases Table 5.3 and Table 5.4 are counterparts of Table 5.1 and Table 5.2, with additional information on the failure ratio and average optimality gaps achieved

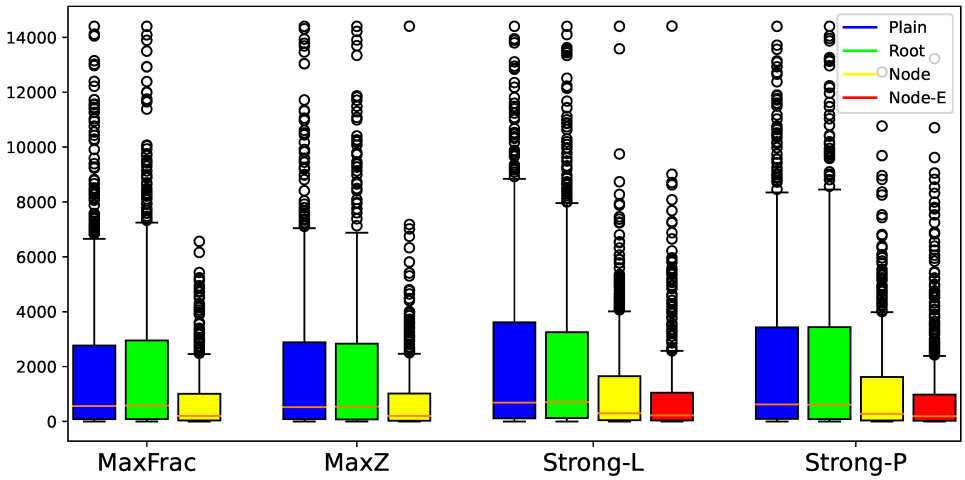

Figure 5-1: Box plot of runtimes for logistic regression problems with scale=1, **easy case**. Results are collected from different choices of λ_0, λ_2, and random seeds of synthetic datasets.

by BnB solvers at time limit. Additional results for synthetic regression datasets can be found in Appendix 5.C.2. We note that Node and Node-E are able to solve more than 10% of the cases that Plain fails to solve within the time limit. Moreoever, the relative optimality gaps obtained using these screening rules are also smaller than those obtained by plain BnB in most cases. On the other hand, Root only achieves very marginal improvements over Plain, which suggest that screening at root only is not as effective as screening within BnB solvers in improving the efficiency of the solver. We further notice that compared to Plain and Root, Node and Node-E are able to explore more nodes and have larger tree sizes than Plain (up to 2 times) within less time. This suggests that the average effective dimensions or the optimization complexities of the subproblems explored by Node and Node-E is relatively low, due to the benefits of screening at each node within BnB trees. Interestingly, Node-E exhibits a smaller average tree size compared to Node, which seems to contradict our theory on exploring more node due to efficiency. Figure 5-2 provides a more detailed and illustrative comparison of distributions of tree sizes on two datasets. We observe that for the cases where Node-E succeeds, search trees of Node-E are substantially smaller than those of Node; for the cases where Node-E also fails, search tree sizes

are similar, and Node and Node-E have slightly larger sizes. This explains why on average Node-E has a smaller tree size.

Table 5.3: Average runtime and tree sizes for **hard case** of logistic regression problems. Results are averaged over different choices of λ_0, λ_2, and random seeds of synthetic datasets. The numbers in the bracket below "scale" indicate the number of hard cases and total cases, respectively. Succ refers to the proportion of successfully solved problems; Gap refers to the relative optimality gap. Time and Size definitions are given in the caption of Table 5.1. The time limit here is 4 hours.

Rules		**MaxFrac**			**MaxZ**			**Strong-L**				**Strong-P**			
		Plain	Root	Node	Plain	Root	Node	Plain	Root	Node	Node-E	Plain	Root	Node	Node-E
scale=1 (258/1000)	Time	14400	14383	13694	14400	14387	13666	14400	14400	14096	13972	14400	14400	14102	13937
	Size	8560	8590	13214	8598	8492	13460	6024	6194	8909	7860	6214	6059	9065	7561
	Succ	0.0	0.008	0.124	0.0	0.012	0.124	0.0	0.0	0.088	0.084	0.0	0.0	0.084	0.08
	Gap	0.118	0.118	0.112	0.125	0.125	0.119	0.129	0.129	0.123	0.123	0.128	0.129	0.123	0.124
scale=3 (117/1000)	Time	14400	14400	13790	14400	14374	13728	14400	14400	14041	14107	14400	14398	13992	14074
	Size	5627	5618	9608	5385	5439	9789	4525	4592	7315	7080	4597	4609	7380	6914
	Succ	0.0	0.0	0.11	0.0	0.017	0.102	0.0	0.0	0.076	0.076	0.0	0.017	0.076	0.067
	Gap	0.108	0.108	0.103	0.119	0.119	0.114	0.113	0.113	0.108	0.108	0.113	0.112	0.108	0.108
scale=5 (92/1000)	Time	14400	14400	13886	14400	14396	13830	14400	14400	14159	14089	14400	14400	14123	14060
	Size	4638	4836	7650	4575	4601	8314	3865	3962	6791	6320	4124	3924	6471	6198
	Succ	0.0	0.0	0.097	0.0	0.011	0.097	0.0	0.0	0.064	0.075	0.0	0.0	0.054	0.064
	Gap	0.119	0.119	0.114	0.134	0.134	0.129	0.124	0.124	0.12	0.12	0.124	0.124	0.12	0.12

In summary, our numerical results for both easy and hard cases demonstrate that node and enhanced node screening strategies can effectively improve the efficiency of the BnB solver for sparse learning problems, while screening at root level (Root) only provides limited improvement. Enhanced node screening (Node-E) is the most efficient screening rule, as it can fix more variables, leading to a reduction in the size of the BnB tree and further reducing the overall running time. Our findings suggest that our proposed screening rules can be used to solve large-scale sparse learning problems in practice effectively.

5.7 Conclusion

We have proposed a novel screening procedure to safely fix relaxed variables to 0 or 1 at each node within a specialized Branch-and-Bound (BnB) solver for sparse learning problems with a generic continuously differentiable convex function and $\ell_0 - \ell_2$ regularization. By establishing optimality conditions and dual bounds for the node relaxation subproblems, we have developed an effective screening procedure at each

Table 5.4: Average runtime and tree sizes for **hard case** of real-world problems. Results are averaged over different choices of λ_0, λ_2. The time limit here is 8 hours. Other details are given in the caption of Table 5.3.

Rules		MaxFrac			MaxZ			Strong-L				Strong-P			
		Plain	Root	Node	Plain	Root	Node	Plain	Root	Node	Node-E	Plain	Root	Node	Node-E
REJA (49/100)	Time	28800	28800	26118	28800	28800	26960	28800	28800	26890	22644	28800	28800	26759	22952
	Size	8503	7757	14979	8697	7549	13835	7155	6750	10970	8852	7060	6442	11569	7772
	Succ	0.0	0.0	0.184	0.0	0.0	0.163	0.0	0.0	0.163	0.326	0.0	0.0	0.163	0.306
	Gap	0.084	0.084	0.074	0.092	0.093	0.083	0.089	0.089	0.079	0.072	0.088	0.089	0.079	0.074
Dexter (45/100)	Time	28800	28732	27249	28800	28800	27590	28800	28800	27308	24799	28800	28800	27412	24803
	Size	6931	7229	12355	6249	6491	11148	6110	6261	9143	7463	5909	5794	9091	7605
	Succ	0.0	0.022	0.11	0.0	0.022	0.088	0.0	0.0	0.133	0.221	0.0	0.022	0.133	0.221
	Gap	0.088	0.088	0.082	0.097	0.096	0.092	0.094	0.093	0.087	0.083	0.094	0.094	0.088	0.083
Arcene (46/100)	Time	28800	28800	27785	28800	28800	28025	28800	28800	27902	25483	28800	28800	27884	25153
	Size	7708	7259	13418	7061	6860	11625	6836	6298	9308	7662	6155	6359	10301	8223
	Succ	0.0	0.0	0.086	0.0	0.0	0.086	0.0	0.0	0.108	0.195	0.0	0.0	0.086	0.239
	Gap	0.086	0.087	0.08	0.097	0.096	0.089	0.091	0.092	0.084	0.079	0.093	0.093	0.086	0.079
Crime (3/100)	Time	28800	28800	28799	28799	28800	28800	28800	28800	28800	28800	28800	28800	28800	28800
	Size	63888	63045	67750	65079	63805	68497	57851	58030	66013	57281	58366	58843	68572	61302
	Succ	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	Gap	0.014	0.014	0.014	0.014	0.014	0.014	0.012	0.012	0.012	0.012	0.012	0.012	0.012	0.012

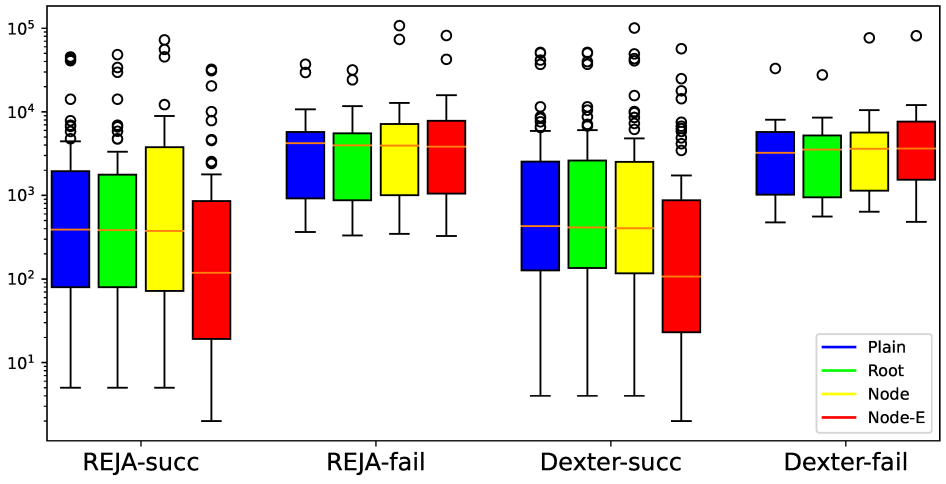

Figure 5-2: Box plot of tree sizes (in log-scale) for **hard cases** of REJA and Dexter. Here, the suffix "succ" and "fail" correspond to the cases where Node-E succeeds or fails to terminate within the 8-hour time limit, respectively.

node within the BnB tree to reduce the optimization cost of subproblems significantly, thus reducing the overall runtime of the solver. Our numerical results demonstrate the effectiveness of our proposed screening rules on both synthetic and real-world datasets. Furthermore, we have introduced an enhanced screening procedure for strong branching rules, which can substantially reduce the size of search trees and further improve the efficiency of the solver. Overall, our proposed screening procedure provides a powerful tool for solving sparse learning problems with $\ell_0 - \ell_2$ regularization.

5.A Additional proofs and examples

5.A.1 Proof of Strong Duality

In this section, we present the proof of strong duality result in Proposition 5.1, and provide the proof sketch of results for node relaxations.

5.A.1.1 Notations

we denote by $\chi\{x \in A\}$ the characteristic function of A, i.e. if $x \in A$, $\chi\{x \in A\} = 0$; otherwise, $\chi\{x \in A\} = \infty$. Similarly $\chi\{a(x) \leq b\}$ denotes the characteristic function of the constraint $a(x) \leq b$. For a function φ, We use $\operatorname{dom}\varphi$ to denote the domain of φ, i.e. $\operatorname{dom}\varphi = \{x : \varphi(x) < \infty\}$. For a set A, we use $\operatorname{ri} A$ to denote the relative interior of A.

5.A.1.2 Proof of Proposition 5.1

By Fenchel Duality Theorem, we have the following lemma:

Lemma 5.2. *Let $\varphi_0, \varphi_1, \varphi_2$ be proper lower-semicontinuous convex functions. If $\operatorname{ri}(\operatorname{dom}\varphi_0) \cap \operatorname{ri}(\operatorname{dom}\varphi_1) \cap \operatorname{ri}(\operatorname{dom}\varphi_2) \neq \varnothing$, then*

$$\begin{aligned}\inf_x \ \varphi_0(x) + \varphi_1(x) + \varphi_2(x) &= \inf_x \sup_{y_1,y_2} \ \varphi_0(x) + x^\top y_1 - \varphi_1^*(y_2) + x^\top y_2 - \varphi_2^*(y_2) \\ &= \sup_{y_1,y_2} \inf_x \ \varphi_0(x) + x^\top y_1 - \varphi_1^*(y_2) + x^\top y_2 - \varphi_2^*(y_2),\end{aligned}$$

where φ_i^* ($i \in \{1,2\}$) is the Fenchel conjugate of φ_i. The ri *can be omitted for* $\operatorname{dom}\varphi_i$ *if φ_i is a polyhedron (for any i).*

Proof. Since φ_1 and φ_2 are convex and lower semi-continuous, we have $\varphi_i(x) = \varphi_i^{**}(x) = \sup_{y_i} \; x^\top y_i - \varphi_i^*(y_i)$, and therefore

$$\inf_x \; \varphi_0(x) + \varphi_1(x) + \varphi_2(x) = \inf_x \sup_{y_1,y_2} \; f(x) + x^\top y_1 - \varphi_1^*(y_2) + x^\top y_2 - \varphi_2^*(y_2).$$

Hence,

$$\begin{aligned}
& \inf_x \; \varphi_0(x) + \varphi_1(x) + \varphi_2(x) \\
\overset{(a)}{=} \; & \sup_{y_2} \; -(\varphi_0 + \varphi_1)^*(-y_2) - \varphi_2^*(y_2) \\
\overset{(b)}{=} \; & \sup_{y_2} \inf_x \varphi_0(x) + \varphi_1(x) + x^\top y_2 - \varphi_2^*(y_2) \\
\overset{(c)}{=} \; & \sup_{y_2} \sup_{y_1} -(\bar{\varphi}_{y_2})^*(-y_1) - \varphi_1^*(y_1) - \varphi_2^*(y_2) \\
\overset{(d)}{=} \; & \sup_{y_1,y_2} \inf_x \; \varphi_0(x) + x^\top y_1 - \varphi_1^*(y_1) + x^\top y_2 - \varphi_2^*(y_2),
\end{aligned}$$

where (a) follows from Fenchel Duality Theorem applied to $\varphi_0 + \varphi_1$ and φ_2; (b) is due to the definition of Fenchel conjugate; (c) follows from Fenchel Duality Theorem applied to $\bar{\varphi}_{y_2}(\cdot) = \varphi_0(\cdot) + \cdot^\top y_2$ and φ_1; and (d) is due to the definition of Fenchel conjugate. Here, the regularity conditions for two applications of Fenchel Duality Theorem are justified by $\operatorname{ri}(\operatorname{dom}\varphi_0) \cap \operatorname{ri}(\operatorname{dom}\varphi_1) \cap \operatorname{ri}(\operatorname{dom}\varphi_2) \neq \varnothing$. □

Based on Lemma 5.2, the road map of the proof of Proposition 5.1 is as follows: let

$$\rho_r(\beta, z) = \frac{\beta^2}{z} + \chi\{z \geq 0\}, \quad \text{and} \quad \iota_r(\beta, z) = \chi\{|\beta| \leq Mz\}, \tag{5.47}$$

and consider the following φ_0, φ_1 and φ_2:

$$\varphi_0(x) = f(\boldsymbol{\beta}) + \lambda_0 \sum_{i=1}^{p} z_i + \chi\{\boldsymbol{z} \in \mathcal{Z}_r\} \tag{5.48a}$$

$$\varphi_1(x) = \lambda_2 \sum_{i=1}^{p} \rho_r(\beta_i, z_i) \tag{5.48b}$$

$$\varphi_2(x) = \sum_{i=1}^{p} \iota_r(\beta_i, z_i). \tag{5.48c}$$

Then, we will show that

$$F(\boldsymbol{\beta}, \boldsymbol{z}) + \chi\{(\boldsymbol{\beta}, \boldsymbol{z}) \in \mathcal{C}_M\} = \max_{\boldsymbol{u},\boldsymbol{v}} L(\boldsymbol{\beta}, \boldsymbol{z}; \boldsymbol{u}, \boldsymbol{v}) \ \text{ for any } \boldsymbol{z} \geq 0 \tag{5.49a}$$

$$F(\boldsymbol{\beta}, \boldsymbol{z}) + \chi\{(\boldsymbol{\beta}, \boldsymbol{z}) \in \mathcal{C}_M, \boldsymbol{z} \in \mathcal{Z}_r\} = \varphi_0(x) + \varphi_1(x) + \varphi_2(x) \tag{5.49b}$$

$$\max_{\boldsymbol{u},\boldsymbol{v}} \min_{\boldsymbol{z}\in\mathcal{Z}_r,\boldsymbol{\beta}} L(\boldsymbol{\beta}, \boldsymbol{z}; \boldsymbol{u}, \boldsymbol{v}) \geq \sup_{y_1,y_2} \inf_{x} \ \varphi_0(x) + x^\top y_1 - \varphi_1^*(y_1) + x^\top y_2 - \varphi_2^*(y_2) \tag{5.49c}$$

Here, (5.49a) implies the first equality in (5.4); (5.49b) and (5.49c), combined with Lemma 5.2, indicate the second equality in (5.4).

The formal proof is as follows:

Proof of Proposition 5.1. We first prove (5.49):

Proof of (5.49a)**.** Recall that for any $z_i \geq 0$, the identity (5.2) holds. It suffices to show for any $z \geq 0, \beta \in \mathbb{R}$, the following holds

$$\chi\{|\beta| \leq Mz\} = \max_{v} \ v\beta - M|v|z. \tag{5.50}$$

This can be checked case by case (LHS/RHS denote left/right hand side of the above equation, resp.):

- $|\beta| \leq Mz$: $LHS = 0 = RHS$ with $v^* = 0$
- $|\beta| > Mz$: $LHS = \infty = RHS$ with $v^* = \infty \operatorname{sign}(\beta)$

Note that the above hold true even if $M = \infty$ with the convention $0 \cdot \infty = 0$.

Proof of (5.49b)**.** This part is obvious by definition of φ_i's.

Proof of (5.49c)**.** To show this, let us first compute ρ_r^* and ι_r^*:

$$\begin{aligned}\rho_r^*(u,u') &= \max_{\beta,z}\ u\beta + u'z - \frac{\beta^2}{z} - \chi\{z \geq 0\} \\ &= \max\left\{0, \max_{z>0,\beta} -\frac{1}{4z}(uz+2\beta)^2 + \left(u' + \frac{u^2}{4}\right)z\right\} \\ &= \chi\left\{u' + \frac{u^2}{4} \leq 0\right\},\end{aligned}$$

where the second equation separates the case of $z = 0$ and $z > 0$.

$$\begin{aligned}\iota_r^*(v,v') &= \max_{\beta,z}\ v\beta + v'z - \chi\{|\beta| \leq Mz\} \\ &= \max_{\beta,z}\ (v\beta - M|v|z) + (v' + M|v|)z + \chi\{|\beta| \leq Mz\} \\ &= \chi\{v' + M|v| \leq 0\}.\end{aligned}$$

Note that if $M = \infty$, this reduces to $\iota_r^*(v,v') = \chi\{v = 0, v' \leq 0\}$.

Therefore, using the above conjugate functions and the decomposability of φ_1 into ρ_r's and that of φ_2 into ι_r's, we have

$$\begin{aligned}&\varphi_0(x) + x^\top y_1 - \varphi_1^*(y_1) + x^\top y_2 - \varphi_2^*(y) \qquad (5.51)\\ &= f(\boldsymbol{\beta}) + \sum_{i=1}^p \left(\lambda_0 z_i + \lambda_2 u_i \beta_i + \lambda_2 u_i' z_i + v_i \beta_i + v_i' z_i\right) + \chi\{\boldsymbol{z} \in \mathcal{Z}_r\} \\ &\qquad - \sum_{i=1}^p \left(\chi\left\{u_i' + \frac{u_i^2}{4} \leq 0\right\} + \chi\{v_i' + M|v_i| \leq 0\}\right).\end{aligned}$$

When $M < \infty$, it is clear to see that when $u_i' = -\frac{u_i^2}{4}$ and $v_i' = -M|v_i|$, the RHS of (5.51) recovers (5.3). Also, by the indicator functions in (5.51), these choices are the upper bounds for u_i' and v_i. Therefore, the objective in (5.51) is upper bounded by L in (5.3), which implies (5.49c). The case for $M = \infty$ can be proved in a similar argument.

Combining (5.49b), (5.49c) and Lemma 5.2, we have

$$\begin{aligned}
&\min_{(\boldsymbol{\beta},\boldsymbol{z})\in\mathcal{C}_M,\boldsymbol{z}\in\mathcal{Z}_r} F(\boldsymbol{\beta},\boldsymbol{z}) \\
=\ &\inf_x\ \varphi_0(x)+\varphi_1(x)+\varphi_2(x) \\
=\ &\sup_{y_1,y_2}\inf_x\ \varphi_0(x)+x^\top y_1-\varphi_1^*(y_1)+x^\top y_2-\varphi_2^*(y_2) \\
\le &\max_{\boldsymbol{u},\boldsymbol{v}}\ \min_{\boldsymbol{\beta}\in\mathbb{R}^p,\boldsymbol{z}\in\mathcal{Z}_r}\ L(\boldsymbol{\beta},\boldsymbol{z};\boldsymbol{u},\boldsymbol{v}).
\end{aligned}$$

By weak duality, we have the other direction of inequality holds, and thus the strong duality holds.

The KKT conditions in (5.5) follow directly from the first-order conditions for the minimization and maximization problems in (5.4). □

5.A.1.3 Proof sketch of strong duality for node relaxation

The node relaxation strong duality proof is similar by replacing $\mathcal{Z}_r$ with $\mathcal{Z}_v$ and enforcing $z_i = 1$ for $i \in \mathcal{F}_1$, $z_i = 0$ for $i \in \mathcal{F}_0$.

To be more specific, let $\rho_1(\beta, z) = \beta^2$, $\iota_1(\beta, z) = \chi\{|\beta| \le M\}$. Then, $\rho_1^*(u, u') = \frac{u^2}{4}, \iota_1^*(v, v') = M|v|$. Note that if $M = \infty$, this is equivalent to $\iota_1^*(v, v') = \chi\{v = 0\}$.

We can then consider

$$\varphi_0(x) = f(\boldsymbol{\beta}) + \lambda_0 \sum_{i\in\mathcal{R}} z_i + \lambda_0|\mathcal{F}_1| + \chi\{\boldsymbol{z} \in \mathcal{Z}_v\} \tag{5.52a}$$

$$\varphi_1(x) = \lambda_2 \sum_{i\in\mathcal{R}} \rho_r(\beta_i, z_i) + \lambda_2 \sum_{i\in\mathcal{F}_1} \rho_1(\beta_i, z_i) \tag{5.52b}$$

$$\varphi_2(x) = \sum_{i\in\mathcal{R}} \iota_r(\beta_i, z_i) + \sum_{i\in\mathcal{F}_1} \iota_1(\beta_i, z_i) \tag{5.52c}$$

Following a similar road map, one can get the strong duality as well as optimality conditions for node relaxation problems.

5.A.1.4 Proof sketch of Proposition 5.4

Similar to the proof for Proposition 5.3, one can get similar equation to (5.15), i.e.

$$L_v(\boldsymbol{\beta}, \boldsymbol{z}; \tilde{\boldsymbol{u}}, \tilde{\boldsymbol{v}}) = f(\boldsymbol{\beta}) + \langle \lambda_2 \tilde{\boldsymbol{u}} + \tilde{\boldsymbol{v}}, \boldsymbol{\beta} \rangle + \sum_{j=1}^{p} \tilde{\Delta}_j z_j. \tag{5.53}$$

At node relaxation, we have for $j \in \mathcal{F}_1$, $z_j \equiv 1$; for $j \in \mathcal{F}_0$, $z_j \equiv 0$. Therefore, this leads to

$$\min_{\boldsymbol{\beta} \in \mathbb{R}^p, \boldsymbol{z} \in \mathcal{Z}_v} L_v(\boldsymbol{\beta}, \boldsymbol{z}; \tilde{\boldsymbol{u}}, \tilde{\boldsymbol{v}}) = f(\tilde{\boldsymbol{\beta}}) - \langle \nabla f(\tilde{\boldsymbol{\beta}}), \tilde{\boldsymbol{\beta}} \rangle - \sum_{j \in \mathcal{R}} (-\tilde{\Delta}_j)_+ + \sum_{j \in \mathcal{F}_1} \tilde{\Delta}_j. \tag{5.54}$$

Following similar arguments in the proof of Proposition 5.3 to optimizing the choices of $\tilde{\Delta}_j$, we obtain the results in Proposition 5.4.

5.B Additional Technical Details

5.B.1 Relationship to [9, 70]

5.B.1.1 Proof sketch of Proposition 5.6

Recall that in Section 5.5, we use $\boldsymbol{X}$ and $\boldsymbol{y}$ to denote the data matrix and the response vector. The sparse regression setting in [9] is equivalent to taking $f(\boldsymbol{\beta}) = \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{\beta}\|^2$, $\lambda_0 = \mu$, $\lambda_2 = \frac{1}{\gamma}$, and $M = \infty$. The sparse logistic regression setting in [70] is equivalent to taking $f(\boldsymbol{\beta}) = \sum_{i=1}^{n} \log(1 + \exp(-y_i \boldsymbol{x}_i^\top \boldsymbol{\beta}))$, $\lambda_0 = \mu$, $\lambda_2 = \frac{1}{\gamma}$ and $M = \infty$. It is not hard to verify that the screening criterion given in Corollary 5.1 exactly recovers that in [9, 70].

5.B.1.2 Examples of screening when $\hat{z} \in (0, 1)$

Consider the following problem

$$\min_{\beta} \ \frac{1}{2}(y - \beta)^2 + \lambda_0 \mathbf{1}\{\beta \neq 0\} + \lambda_2 \beta^2. \tag{5.55}$$

Given any β, we have

$$\underline{F}_r = \frac{1}{2}(y-\beta)^2 - \beta(\beta - y) = \frac{1}{2}y^2 - \frac{1}{2}\beta^2 \tag{5.56}$$

$$\Delta = \lambda_0 - \frac{(\beta - y)^2}{4\lambda_2} \tag{5.57}$$

$$z = \min\{1, \max\{0, |\beta|\sqrt{\lambda_2/\lambda_0}\}\} \tag{5.58}$$

Therefore, in this case

$$\underline{F}_r(z=0) = \frac{1}{2}y^2 - \frac{1}{2}\beta^2 \tag{5.59}$$

$$\underline{F}_r(z=1) = \frac{1}{2}y^2 - \frac{1}{2}\beta^2 + \lambda_0 - \frac{(\beta - y)^2}{4\lambda_2}. \tag{5.60}$$

Screening to $z^\dagger = 0$. Taking $y = 3, \lambda_0 = 100, \lambda_2 = 10^{-2}, M = \infty$, we have the optimal solutions to the MIP and the relaxation problems are

$$(\beta^\dagger, z^\dagger, F^\dagger) = (0, 0, 4.5), \quad \text{and} \quad (\beta^*, z^*, F^*) = (1, 0.01, 4) \tag{5.61}$$

Let $\bar{F}_I = F^\dagger$. Since $z^* = 0.01 \in (0, 1)$, no screening happens at the optimal relaxation solution (β^*, z^*).

For a general β, $\underline{F}_r(z=0) = 4.5 - 0.5\beta^2 \leq \bar{F}_I = 4.5$, but $\underline{F}_r(z=1) = -25.5\beta^2 + 150\beta - 120.5 > \bar{F}_I = 4.5$ when

$$\beta \in \left(\frac{50}{17} - \frac{5\sqrt{390}}{51}, \frac{50}{17} + \frac{5\sqrt{390}}{51}\right) \approx (1.00506, 4.87730). \tag{5.62}$$

This corresponds to

$$z = |\beta|/100 \in (0.01005, 0.04877) \subset (0, 1). \tag{5.63}$$

Screening to $z^\dagger = 1$. Taking $y = 1, \lambda_0 = \lambda_2 = 0.05, M = \infty$, we have the optimal

solutions to the MIP and the relaxation problems are

$$(\beta^\dagger, z^\dagger, F^\dagger) = (\frac{10}{11}, 1, \frac{21}{220}), \quad \text{and} \quad (\beta^*, z^*, F^*) = (\frac{9}{10}, \frac{9}{10}, \frac{19}{200}) \tag{5.64}$$

Let $\bar{F}_I = F^\dagger$. Since $z^* = 0.9 \in (0, 1)$, no screening happens at the optimal relaxation solution (β^*, z^*).

For a general β, $\underline{F}_r(z = 1) = -\frac{11}{2}(\beta - \frac{10}{11})^2 + \frac{21}{220} \leq \bar{F}_I$, but $\underline{F}_r(z = 0) = 0.5 - 0.5\beta^2 > \bar{F}_I = \frac{21}{220}$ when

$$\beta \in \left(-\sqrt{\frac{89}{110}}, \sqrt{\frac{89}{110}}\right) \approx (-0.89949, 0.89949). \tag{5.65}$$

This corresponds to

$$z = |\beta| \in [0, 0.89949) \subseteq [0, 1). \tag{5.66}$$

5.B.2 Relationship to [105]

The following proposition suggests that the node screening test presented in [105] is a special case of our rule presented in Proposition 5.7.

Proposition 5.8. *Proposition 5.7 recovers the screening tests presented in [105] when* $\lambda_2 = 0$ *in the case of linear regression.*

The sparse regression setting in [105] is equivalent to taking $f(\boldsymbol{\beta}) = \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|^2$, $\lambda_0 = \lambda$ and $\lambda_2 = 0$, and it is not hard to verify that our screening rules presented in Corollary 5.2 exactly recovers [105].

5.C Additional Experiment Details

5.C.1 Additional experiment setup

Following [120], we adopt the similar parameters for the BnB solver. Specifically, relative optimality gap for BnB solver termination is set to 1%, integer feasibility

tolerance is set to 10^{-4}, and primal-dual optimality gap for subproblem solver is set to 10^{-8} (as opposed to 10^{-5} in [120]).

5.C.1.1 Synthetic data generation

For classification datasets, we take $\rho = 0.05$. For the regression model, the data matrix $\boldsymbol{X} \in \mathbb{R}^{n \times p}$ is generated in the same way as classification model with constant correlation $\rho = 0.15$. The response vector $\boldsymbol{y}$ is obtained from the linear model $\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta}^c + \boldsymbol{\epsilon}$, where the noise vector ϵ with independent and identically distributed (i.i.d.) coordinates $\epsilon_j \sim N(0, \sigma^2)$ for $j \in [n]$. Here, the standard error σ is determined by so-called signal-to-noise ratio (SNR), defined as $\text{SNR} = Var(\boldsymbol{X}\boldsymbol{\beta}^c)/\sigma^2$. We consider $\text{SNR} \in \{1, 1.5, 2\}$ in the experiments, and like classification datasets, for each SNR value, we generate 10 random data sets.

5.C.1.2 Real-world data details

- **REJA**: this dataset is taken from `https://archive.ics.uci.edu/ml/datasets/REJAFADA+`. We consider a training dataset with $n = 1996$ and $p = 477$.

- **Dexter**: this dataset is taken from `https://archive.ics.uci.edu/ml/datasets/dexter`. We consider a training dataset with $n = 300$ and $p = 457$.

- **Arcene**: this dataset is taken from `https://archive.ics.uci.edu/ml/datasets/Arcene`. We consider a training dataset with $n = 100$ and $p = 482$.

- **Crime**: this dataset is taken from `https://archive.ics.uci.edu/ml/datasets/communities+and+crime`. We remove all features with missing values and consider a training dataset with $n = 1999$ and $p = 94$.

5.C.1.3 Choices of λ_0, λ_2

Synthetic datasets To determine the grid of regularization parameters (λ_0, λ_2), we follow these steps.

1. We first create a grid of $\tilde{\lambda}_2$ in a wide range. For the regression model, we take 10 values equispaced on a logarithmic scale in the range $[10^1, 10^{1.25}]$, while for the classification model, we take 10 points equispaced on a logarithmic scale in the range $[10^{-0.5}, 10^{1.5}]$.

2. For each $\tilde{\lambda}_2$, we compute a regularization path of $\tilde{\lambda}_0$: $\tilde{\lambda}_0^1(\tilde{\lambda}_2) > \tilde{\lambda}_0^2(\tilde{\lambda}_2) > \cdots > \tilde{\lambda}_0^m(\tilde{\lambda}_2)$. We use adaptive selection rules as described in [116] to generate a sequence $\tilde{\lambda}_0^i$ that avoids duplicate solutions. For each $(\tilde{\lambda}_0, \tilde{\lambda}_2)$, we use our plain BnB solver to approximately solve the problem, with maximal tree depth set to 5 and a time limit of 30 seconds. We denote by $\hat{\beta}(\tilde{\lambda}_0^i(\tilde{\lambda}_2), \tilde{\lambda}_2)$ the solution obtained from Plain.

3. Determine $\lambda_0^*(\tilde{\lambda}_2)$ that minimizes the ℓ_2 estimation error of $\hat{\boldsymbol{\beta}}$. For the regression model, we choose

$$\lambda_0^*(\tilde{\lambda}_2) \in \arg\min_{\tilde{\lambda}_0^i(\tilde{\lambda}_2)} \|\boldsymbol{\beta}^c - \hat{\boldsymbol{\beta}}(\tilde{\lambda}_0^i(\tilde{\lambda}_2), \tilde{\lambda}_2)\|_2, \tag{5.67}$$

   and for the classification model, we choose

$$\lambda_0^*(\tilde{\lambda}_2) \in \arg\min_{\tilde{\lambda}_0^i(\tilde{\lambda}_2)} \|s\boldsymbol{\beta}^c - \hat{\boldsymbol{\beta}}(\tilde{\lambda}_0^i(\tilde{\lambda}_2), \tilde{\lambda}_2)\|_2. \tag{5.68}$$

4. Choose the λ_0 grid for each $\tilde{\lambda}_2$. For the regression model, we take 10 values equispaced on a linear scale in the range $[1.25\lambda_0^*(\tilde{\lambda}_2), 1.5\lambda_0^*(\tilde{\lambda}_2)]$. For the classification model, we take 10 values equispaced on a linear scale in the range $[0.5\lambda_0^*(\tilde{\lambda}_2), 2\lambda_0^*(\tilde{\lambda}_2)]$.

Real datasets Since there is no ground truth of $\boldsymbol{\beta}^c$ known to us. Instead, we perform a 5-fold cross-validation using our implementation of `L0learn` [116] to find the optimal regularization parameters λ_0^* and λ_2^*. Then, our experiments are run on the grids where we take 10 values of λ_2 equispaced on a logarithmic scale in the range $\left\{10^{-1/2}\lambda_2^*, 10^{1/2}\lambda_2^*\right\}$ and vary λ_0 to generate solutions of different support sizes.

5.C.2 Additional numerical results

Finally, we present additional numerical results for synthetic regression datasets. Table 5.5, Table 5.6 and Figure 5-3 provide simulation results that correspond to those in Table 5.1, Table 5.3 and Figure 5-1 respectively, but for regression model. The tables and figure reveal a qualitatively very similar story to that presented in Section 5.6.

Table 5.5: Average runtime and tree sizes for **easy case** of synthetic regression problems. Results are averaged over different choices of λ_0, λ_2, and random seeds of synthetic datasets. Details are given in the caption of Table 5.1.

Rules		**MaxFrac**			**MaxZ**			**Strong-L**				**Strong-P**			
		Plain	Root	Node	Plain	Root	Node	Plain	Root	Node	Node-E	Plain	Root	Node	Node-E
SNR=1 (803/1000)	Time	2932	2899	1692	2730	2661	1594	3114	3066	1966	1371	3057	3071	1896	1314
	Size	9048	9035	9098	8719	8720	8750	9020	9021	9085	5372	9161	9139	9191	5234
SNR=1.5 (847/1000)	Time	1613	1598	968	1479	1474	900	1704	1705	1100	602	1716	1712	1068	568
	Size	4885	4853	4916	4751	4751	4788	4867	4819	4977	2258	4899	4889	5037	2182
SNR=2 (1000/1000)	Time	184	181	107	162	160	96	196	193	121	82	189	190	118	78
	Size	624	624	624	602	602	602	625	625	624	367	619	619	618	352

Table 5.6: Average runtime and tree sizes for **hard case** of synthetic regression problems. Results are averaged over different choices of λ_0, λ_2, and random seeds of synthetic datasets. Details are given in the caption of Table 5.3.

Rules		**MaxFrac**			**MaxZ**			**Strong-L**				**Strong-P**			
		Plain	Root	Node	Plain	Root	Node	Plain	Root	Node	Node-E	Plain	Root	Node	Node-E
SNR=1 (197/1000)	Time	14400	14400	14166	14400	14390	14083	14400	14387	14184	14216	14400	14378	14165	14122
	Size	21803	21466	22986	20429	20111	22494	18297	18437	19541	17990	17716	17835	19263	17570
	Succ	0.0	0.0	0.07	0.0	0.01	0.08	0.0	0.01	0.061	0.061	0.0	0.015	0.061	0.075
	Gap	0.063	0.063	0.061	0.065	0.065	0.063	0.064	0.064	0.062	0.062	0.064	0.064	0.063	0.063
SNR=1 (153/1000)	Time	14400	14361	14235	14400	14375	14142	14400	14397	14284	14213	14400	14382	14249	14159
	Size	12022	12308	13449	11611	11823	13501	10305	10421	11579	10732	10503	10055	11857	10782
	Succ	0.0	0.013	0.033	0.0	0.013	0.052	0.0	0.007	0.039	0.033	0.0	0.013	0.033	0.052
	Gap	0.063	0.062	0.061	0.065	0.065	0.063	0.066	0.066	0.064	0.065	0.066	0.066	0.065	0.065

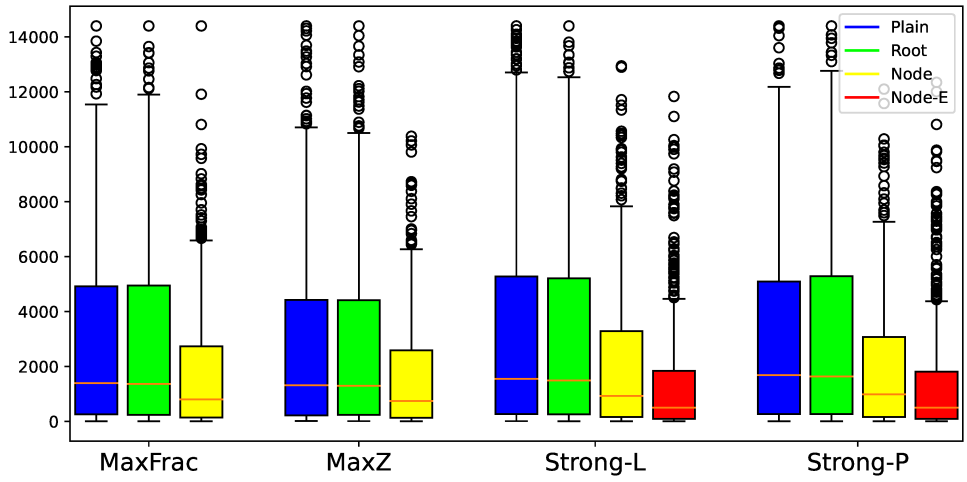

Figure 5-3: Box plot of runtimes for synthetic regression problems with SNR=1, **easy case**. Results are collected from different choices of λ_0, λ_2, and random seeds of synthetic datasets.

Chapter 6

Conclusion

This thesis presented large-scale optimization algorithms for several machine learning problems under structural constraints. In Chapters 2 and 3, we focused on problems in nonparametric statistics with shape constraints and developed efficient convex optimization approaches with novel convergence guarantees. In particular, in Chapter 2, we present an active set algorithmic framework for solving large-scale subgradient regularized convex regression problem, which leverages proximal gradient methods as well as different random and greedy techniques for active set augmentation. We show novel linear convergence guarantees for our algorithms in the absence of strong convexity. In practice, our algorithms can approximately solve the instances with $n = 10^5$ and $d = 10$ within minutes. In Chapter 3, we develop a scalable computational framework for computing log-concave MLE, based on randomized smoothing and Nesterov smoothing, combined with an integral discretization of increasing grid sizes. We provide the convergence guarantees and our new framework is up to 30 times faster than the existing convex methods. Our framework can also apply to other shape constrained density estimation problems. In Chapters 4 and 5, we switched the gear to investigate into sparse learning problems and proposed scalable discrete optimization methods. Specifically, in Chapter 4, we propose a novel estimator for sparse precision matrix for iid multivariate Gaussian samples, based on $\ell_0\ell_2$-regularized pseudolikelihood. We provide statistical guarantees for our proposal in terms of estimation and variable selection. We further reformulate the problem into

MIP, and develop a fast approximate algorithm as well as a scalable exact algorithm (specialized BnB solver) for the problem. Our exact algorithm is computationally scalable to $p \approx 10{,}000$ (corresponding to 50 million parameters). In Chapter 5, we design a safe screening procedure at each node within BnB solver for a general $\ell_0\ell_2$-regularized sparse learning problem. We demonstrate that the deployment of such screening procedure can improve the runtime of BnB algorithm up to 2 times. When using strong branching strategies, it can further reduce the tree sizes and lead to a substantial runtime improvement.

There are multiple exciting directions for future work in sparse learning and convex optimization. The $\ell_0\ell_2$ penalty has a drawback that it can control the exact sparsity of the final solution — for example, if one wishes to select k variables out of p, sometimes it is hard to find a hyperparameter grid (λ_0, λ_2) so that the final solution has exactly k nonzeros. Hence one interesting direction could be replacing ℓ_0 penalty with cardinality constraint, i.e. $\|\boldsymbol{\beta}\|_0 \leq k$, for sparse learning problem, and developing scalable approximate and exact algorithms.

The methodological work presented in Chapters 2 to 5 has many applications and generalizations in areas of finance and neural networks. In [125][1], we develop a `GregNets` framework for joint learning time series forecasting models and partial correlation structures that leverages graph connectivity from financial knowledge graphs (based on pseudo-likelihood). In [51][2], we propose a flexible optimization framework to simultaneously learn covariance matrices across different time periods under suitable structural assumptions on the period-specific covariance matrices and time-varying regularizers. In [23][3], we propose an optimization framework `CHITA` for neural network pruning/sparsification based on an ℓ_0 cardinality-constrained sparse regression problem.

[1]This is a joint work with Shibal Ibrahim, Yada Zhu, Pin-Yu Chen, Yang Zhang and Rahul Mazumder.

[2]This is a joint work with Riade Benbaki, Yada Zhu and Rahul Mazumder.

[3]A workshop version of this appears in [50]. This is a joint work with Riade Benbaki, Xiang Meng, Hussein Hazimeh, Natalia Ponomareva, Zhe Zhao, and Rahul Mazumder.

Bibliography

[1] Tobias Achterberg. Constraint integer programming. PhD thesis, Technische Universitat Berlin, 2007.

[2] Tobias Achterberg, Thorsten Koch, and Alexander Martin. Branching rules revisited. Operations Research Letters, 33(1):42–54, 2005.

[3] Alekh Agarwal, Peter L Bartlett, Pradeep Ravikumar, and Martin J Wainwright. Information-theoretic lower bounds on the oracle complexity of stochastic convex optimization. IEEE Transactions on Information Theory, 5(58):3235–3249, 2012.

[4] M Selim Aktürk, Alper Atamtürk, and Sinan Gürel. A strong conic quadratic reformulation for machine-job assignment with controllable processing times. Operations Research Letters, 37(3):187–191, 2009.

[5] David Applegate, Robert Bixby, Vašek Chvátal, and William Cook. On the solution of traveling salesman problems. Documenta Mathematica, pages 645–656, 1998.

[6] MOSEK ApS. The MOSEK optimization toolbox for Python manual. Version 9.3., 2022.

[7] Aleksandr Y Aravkin, Anju Kambadur, Aurélie C Lozano, and Ronny Luss. Sparse quantile huber regression for efficient and robust estimation. arXiv preprint arXiv:1402.4624, 2014.

[8] Alper Atamturk and Andres Gomez. Rank-one convexification for sparse regression. arXiv preprint arXiv:1901.10334, 2019.

[9] Alper Atamturk and Andrés Gómez. Safe screening rules for ℓ_0-regression from perspective relaxations. In International conference on machine learning, pages 421–430. PMLR, 2020.

[10] Brian Axelrod, Ilias Diakonikolas, Alistair Stewart, Anastasios Sidiropoulos, and Gregory Valiant. A polynomial time algorithm for log-concave maximum likelihood via locally exponential families. In Advances in Neural Information Processing Systems, pages 7723–7735, 2019.

[11] Necdet Serhat Aybat and Zi Wang. A parallelizable dual smoothing method for large scale convex regression problems. arXiv preprint arXiv:1608.02227, 2016.

[12] Kazuoki Azuma. Weighted sums of certain dependent random variables. Tohoku Mathematical Journal, Second Series, 19(3):357–367, 1967.

[13] Fadoua Balabdaoui. Consistent estimation of a convex density at the origin. Mathematical Methods of Statistics, 16(2):77–95, 2007.

[14] Gábor Balázs. Convex regression: Theory, practice, and applications. PhD thesis, University of Alberta, 2016.

[15] C Bradford Barber, David P Dobkin, and Hannu Huhdanpaa. The quickhull algorithm for convex hull. Technical report, Technical Report GCG53, Geometry Center, Univ. of Minnesota, 1993.

[16] Rina Foygel Barber and Richard J Samworth. Local continuity of log-concave projection, with applications to estimation under model misspecification. Bernoulli, to appear, 2021.

[17] Amir Beck. First-order methods in optimization, volume 25. SIAM, 2017.

[18] Amir Beck and Yonina C Eldar. Sparsity constrained nonlinear optimization: Optimality conditions and algorithms. SIAM Journal on Optimization, 23(3):1480–1509, 2013.

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

[20] Kayhan Behdin and Rahul Mazumder. Sparse PCA: A new scalable estimator based on integer programming. arXiv preprint arXiv:2109.11142, 2021.

[21] Pierre C Bellec. Sharp oracle inequalities for least squares estimators in shape restricted regression. The Annals of Statistics, 46(2):745–780, 2018.

[22] Pietro Belotti, Christian Kirches, Sven Leyffer, Jeff Linderoth, James Luedtke, and Ashutosh Mahajan. Mixed-integer nonlinear optimization. Acta Numerica, 22:1–131, 2013.

[23] Riade Benbaki, Wenyu Chen, Xiang Meng, Hussein Hazimeh, Natalia Ponomareva, Zhe Zhao, and Rahul Mazumder. Fast as chita: Neural network pruning with combinatorial optimization. arXiv preprint arXiv:2302.14623, 2023.

[24] Michel Bénichou, Jean-Michel Gauthier, Paul Girodet, Gerard Hentges, Gerard Ribière, and Olivier Vincent. Experiments in mixed-integer linear programming. Mathematical Programming, 1:76–94, 1971.

[25] Dimitri P Bertsekas. Nonlinear programming. Journal of the Operational Research Society, 48(3):334–334, 1997.

[26] Dimitri P Bertsekas. Convex Optimization Theory. Athena Scientific Belmont, 2009.

[27] D.P. Bertsekas. Nonlinear Programming. Athena scientific optimization and computation series. Athena Scientific, 2016.

[28] Dimitris Bertsimas, Angela King, and Rahul Mazumder. Best subset selection via a modern optimization lens. The annals of statistics, 44(2):813–852, 2016.

[29] Dimitris Bertsimas, Jourdain Lamperski, and Jean Pauphilet. Certifiably optimal sparse inverse covariance estimation. Mathematical Programming, 184(1):491–530, 2020.

[30] Dimitris Bertsimas and Nishanth Mundru. Sparse convex regression. INFORMS Journal on Computing, 2020.

[31] Dimitris Bertsimas and Nishanth Mundru. Sparse convex regression. INFORMS Journal on Computing, 33(1):262–279, 2021.

[32] Dimitris Bertsimas and Bart Van Parys. Sparse high-dimensional regression: Exact scalable algorithms and phase transitions. The Annals of Statistics, 48(1):300 – 323, 2020.

[33] Dimitris Bertsimas, Jean Pauphilet, and Bart Van Parys. Sparse regression: Scalable algorithms and empirical performance. Statistical science, 35(4):555–578, 2020.

[34] Dimitris Bertsimas and John N Tsitsiklis. Introduction to linear optimization, volume 6. Athena Scientific Belmont, MA, 1997.

[35] Julian Besag. Statistical analysis of non-lattice data. Journal of the Royal Statistical Society: Series D (The Statistician), 24(3):179–195, 1975.

[36] Thomas Blumensath and Mike E Davies. Iterative hard thresholding for compressed sensing. Applied and computational harmonic analysis, 27(3):265–274, 2009.

[37] Jérôme Bolte, Trong Phong Nguyen, Juan Peypouquet, and Bruce W Suter. From error bounds to the complexity of first-order descent methods for convex functions. Mathematical Programming, 165(2):471–507, 2017.

[38] Pierre Bonami, Jon Lee, Sven Leyffer, and Andreas Wächter. More branch-and-bound experiments in convex nonlinear integer programming. Preprint ANL/MCS-P1949-0911, Argonne National Laboratory, Mathematics and Computer Science Division, 91, 2011.

[39] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, Jonathan Eckstein, et al. Distributed optimization and statistical learning via the alternating direction method of multipliers. Foundations and Trends® in Machine learning, 3(1):1–122, 2011.

[40] Stephen Boyd and Lieven Vandenberghe. Convex Optimization. Cambridge University Press, Cambridge, 2004.

[41] HD Brunk, Richard E Barlow, Daniel J Bartholomew, and James M Bremner. Statistical Inference under Order Restrictions: The Theory and Application of Isotonic Regression. John Wiley & Sons, 1972.

[42] V V Buldygin and Yu V Kozachenko. Metric Characterization of Random Variables and Random Processes, volume 188. American Mathematical Society, 2000.

[43] Florentina Bunea, Alexandre B Tsybakov, and Marten H Wegkamp. Aggregation for gaussian regression. The Annals of Statistics, pages 1674–1697, 2007.

[44] T Tony Cai and Mark G Low. A framework for estimation of convex functions. Statistica Sinica, 25:423–456, 2015.

[45] Tony Cai, Weidong Liu, and Xi Luo. A constrained ell-1 minimization approach to sparse precision matrix estimation. Journal of the American Statistical Association, 106(494):594–607, 2011.

[46] Emmanuel J Candes and Mark A Davenport. How well can we estimate a sparse vector? Applied and Computational Harmonic Analysis, 34(2):317–323, 2013.

[47] Timothy Carpenter, Ilias Diakonikolas, Anastasios Sidiropoulos, and Alistair Stewart. Near-optimal sample complexity bounds for maximum likelihood estimation of multivariate log-concave densities. In Conference On Learning Theory, pages 1234–1262, 2018.

[48] Chih-Chung Chang and Chih-Jen Lin. Libsvm: a library for support vector machines. ACM transactions on intelligent systems and technology (TIST), 2(3):1–27, 2011.

[49] Sabyasachi Chatterjee, Adityanand Guntuboyina, and Bodhisattva Sen. On risk bounds in isotonic and other shape restricted regression problems. The Annals of Statistics, 43(4):1774–1800, 2015.

[50] Wenyu Chen, Riade Benbaki, Xiang Meng, and Rahul Mazumder. Network pruning at scale: A discrete optimization approach. In OPT 2022: Optimization for Machine Learning (NeurIPS 2022 Workshop), 2022.

[51] Wenyu Chen, Riade Benbaki, Yada Zhu, and Rahul Mazumder. Dynamic covariance estimation under structural assumptions via a joint optimization approach. In 2022 KDD Workshop on Machine Learning in Finance, 2022.

[52] Wenyu Chen and Rahul Mazumder. Multivariate convex regression at scale. arXiv preprint arXiv:2005.11588, 2020.

[53] Wenyu Chen, Rahul Mazumder, and Richard J Samworth. A new computational framework for log-concave density estimation. arXiv preprint arXiv:2105.11387, 2021.

[54] Yining Chen and Richard J Samworth. Smoothed log-concave maximum likelihood estimation with applications. Statistica Sinica, 23:1373–1398, 2013.

[55] Yining Chen and Richard J Samworth. Generalized additive and index models with shape constraints. Journal of the Royal Statistical Society, Series B, 78:729–754, 2016.

[56] Frank H Clarke. Optimization and Nonsmooth Analysis, volume 5. SIAM, Philadelphia, 1990.

[57] David R Cox. Regression models and life-tables. Journal of the Royal Statistical Society: Series B (Methodological), 34(2):187–202, 1972.

[58] Alison Cozad, Nikolaos V Sahinidis, and David C Miller. Learning surrogate models for simulation-based optimization. AIChE Journal, 60(6):2211–2227, 2014.

[59] Madeleine Cule, Robert B. Gramacy, and Richard Samworth. LogConcDEAD: An R package for maximum likelihood estimation of a multivariate log-concave density. Journal of Statistical Software, 29(2):1–20, 2009.

[60] Madeleine Cule and Richard Samworth. Theoretical properties of the log-concave maximum likelihood estimator of a multidimensional density. Electronic Journal of Statistics, 4:254–270, 2010.

[61] Madeleine Cule, Richard Samworth, and Michael Stewart. Maximum likelihood estimation of a multi-dimensional log-concave density. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 72(5):545–607, 2010.

[62] Madeleine L Cule and Lutz Dümbgen. On an auxiliary function for log-density estimation. arXiv preprint arXiv:0807.4719, 2008.

[63] Arnak S Dalalyan. Theoretical guarantees for approximate sampling from smooth and log-concave densities. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 79(3):651–676, 2017.

[64] George B Dantzig and Philip Wolfe. Decomposition principle for linear programs. Operations research, 8(1):101–111, 1960.

[65] Gamarnik David and Zadik Ilias. High dimensional regression with binary coefficients. estimating squared error and a phase transtition. In Conference on learning theory, pages 948–953. PMLR, 2017.

[66] Antoine Dedieu, Hussein Hazimeh, and Rahul Mazumder. Learning sparse classifiers: Continuous and mixed integer optimization perspectives. arXiv preprint arXiv:2001.06471, 2020.

[67] Antoine Dedieu, Hussein Hazimeh, and Rahul Mazumder. Learning sparse classifiers: Continuous and mixed integer optimization perspectives. The Journal of Machine Learning Research, 22(1):6008–6054, 2021.

[68] A. P. Dempster. Covariance selection. Biometrics, 28(1):157–175, 1972.

[69] Santanu S Dey, Yatharth Dubey, Marco Molinaro, and Prachi Shah. A theoretical and computational analysis of full strong-branching. arXiv preprint arXiv:2110.10754, 2021.

[70] Anna Deza and Alper Atamturk. Safe screening for logistic regression with $\ell_0 - \ell_2$ regularization. arXiv preprint arXiv:2202.00467, 2022.

[71] Sudhakar Dharmadhikari and Kumar Joag-Dev. Unimodality, Convexity, and Applications. Elsevier, 1988.

[72] Hongbo Dong, Kun Chen, and Jeff Linderoth. Regularization vs. relaxation: A conic optimization perspective of statistical variable selection. arXiv preprint arXiv:1510.06083, 2015.

[73] Charles R Doss and Jon A Wellner. Global rates of convergence of the MLEs of log-concave and s-concave densities. The Annals of Statistics, 44(3):954, 2016.

[74] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017.

[75] John C Duchi, Peter L Bartlett, and Martin J Wainwright. Randomized smoothing for stochastic optimization. SIAM Journal on Optimization, 22(2):674–701, 2012.

[76] Lutz Dümbgen and Kaspar Rufibach. Maximum likelihood estimation of a log-concave density and its distribution function: Basic properties and uniform consistency. Bernoulli, 15(1):40–68, 2009.

[77] Lutz Dümbgen and Kaspar Rufibach. logcondens: Computations related to univariate log-concave density estimation. Journal of Statistical Software, 39:1–28, 2011.

[78] Lutz Dümbgen, Richard Samworth, and Dominic Schuhmacher. Approximation by log-concave distributions, with applications to regression. The Annals of Statistics, 39(2):702–730, 2011.

[79] Lutz Dümbgen, Richard J. Samworth, and Jon A. Wellner. Bounding distributional errors via density ratios. Bernoulli, 27:818–852, 2021.

[80] Marco A Duran and Ignacio E Grossmann. An outer-approximation algorithm for a class of mixed-integer nonlinear programs. Mathematical programming, 36:307–339, 1986.

[81] Cécile Durot and Hendrik P Lopuhaä. Limit theory in monotone function estimation. Statistical Science, 33(4):547–567, 2018.

[82] Jianqing Fan, Yongyi Guo, and Ziwei Zhu. When is best subset selection the" best"? arXiv preprint arXiv:2007.01478, 2020.

[83] Jianqing Fan, Yuan Liao, and Han Liu. An overview of the estimation of large covariance and precision matrices. The Econometrics Journal, 19(1):C1–C32, 2016.

[84] Billy Fang and Adityanand Guntuboyina. On the risk of convex-constrained least squares estimators under misspecification. Bernoulli, 25(3):2206–2244, 2019.

[85] Oliver Y Feng, Adityanand Guntuboyina, Arlene KH Kim, and Richard J Samworth. Adaptation in multivariate log-concave density estimation. The Annals of Statistics, 49:129–153, 2021.

[86] Olivier Fercoq, Alexandre Gramfort, and Joseph Salmon. Mind the duality gap: safer rules for the lasso. In International Conference on Machine Learning, pages 333–342. PMLR, 2015.

[87] Alyson K Fletcher, Sundeep Rangan, and Vivek K Goyal. Necessary and sufficient conditions for sparsity pattern recovery. IEEE Transactions on Information Theory, 55(12):5758–5772, 2009.

[88] Antonio Frangioni and Claudio Gentile. Perspective cuts for a class of convex 0–1 mixed integer programs. Mathematical Programming, 106(2):225–236, 2006.

[89] Jerome Friedman, Trevor Hastie, and Rob Tibshirani. Regularization paths for generalized linear models via coordinate descent. Journal of statistical software, 33(1):1, 2010.

[90] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Sparse inverse covariance estimation with the graphical lasso. Biostatistics, 9(3):432–441, 2008.

[91] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Applications of the lasso and grouped lasso to the estimation of sparse graphical models. Technical report, Technical report, Stanford University, 2010.

[92] Jerome H Friedman. Greedy function approximation: a gradient boosting machine. Annals of statistics, pages 1189–1232, 2001.

[93] Jerome H Friedman. The elements of statistical learning: Data mining, inference, and prediction. springer open, 2017.

[94] Wenjiang J Fu. Penalized regressions: the bridge versus the lasso. Journal of computational and graphical statistics, 7(3):397–416, 1998.

[95] Laurent El Ghaoui, Vivian Viallon, and Tarek Rabbani. Safe feature elimination for the lasso and sparse supervised learning problems. arXiv preprint arXiv:1009.4219, 2010.

[96] Avishek Ghosh, Ashwin Pananjady, Adityanand Guntuboyina, and Kannan Ramchandran. Max-affine regression: Provable, tractable, and near-optimal statistical estimation. arXiv preprint arXiv:1906.09255, 2019.

[97] Andrés Gómez and Oleg A Prokopyev. A mixed-integer fractional optimization approach to best subset selection. INFORMS Journal on Computing, 33(2):551–565, 2021.

[98] Eitan Greenshtein. Best subset selection, persistence in high-dimensional statistical learning and optimization under l_1 constraint. The Annals of Statistics, pages 2367–2386, 2006.

[99] Ulf Grenander. On the theory of mortality measurement: part ii. Scandinavian Actuarial Journal, 1956(2):125–153, 1956.

[100] Piet Groeneboom and Geurt Jongbloed. Nonparametric Estimation under Shape Constraints, volume 38. Cambridge University Press, Cambridge., 2014.

[101] Oktay Günlük and Jeff Linderoth. Perspective reformulations of mixed integer nonlinear programs with indicator variables. Mathematical programming, 124(1):183–205, 2010.

[102] Adityanand Guntuboyina and Bodhisattva Sen. Global risk bounds and adaptation in univariate convex regression. Probability Theory and Related Fields, 163(1-2):379–411, 2015.

[103] Adityanand Guntuboyina and Bodhisattva Sen. Nonparametric shape-restricted regression. Statistical Science, 33(4):568–594, 2018.

[104] Gurobi Optimization, LLC. Gurobi optimizer reference manual, 2021.

[105] Théo Guyard, Cédric Herzet, and Clément Elvira. Node-screening tests for the l_0-penalized least-squares problem. In ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 5448–5452. IEEE, 2022.

[106] William W Hager and Hongchao Zhang. Projection onto a polyhedron that exploits sparsity. SIAM Journal on Optimization, 26(3):1773–1798, 2016.

[107] Qiyang Han. Global empirical risk minimizers with “shape constraints" are rate optimal in general dimensions. The Annals of Statistics, to appear, 2021.

[108] Qiyang Han, Tengyao Wang, Sabyasachi Chatterjee, and Richard J Samworth. Isotonic regression in general dimensions. The Annals of Statistics, 47(5):2440–2471, 2019.

[109] Qiyang Han and Jon A Wellner. Approximation and estimation of s-concave densities via rényi divergences. The Annals of Statistics, 44(3):1332, 2016.

[110] Qiyang Han and Jon A Wellner. Multivariate convex regression: global risk bounds and adaptation. arXiv preprint arXiv:1601.06844, 2016.

[111] Qiyang Han and Jon A Wellner. Convergence rates of least squares regression estimators with heavy-tailed errors. The Annals of Statistics, 47(4):2286–2319, 2019.

[112] Lauren A Hannah and David B Dunson. Multivariate convex regression with adaptive partitioning. The Journal of Machine Learning Research, 14(1):3261–3294, 2013.

[113] Trevor Hastie, Robert Tibshirani, Jerome H Friedman, and Jerome H Friedman. The elements of statistical learning: data mining, inference, and prediction, volume 2. Springer, 2009.

[114] Trevor Hastie, Robert Tibshirani, and Ryan J Tibshirani. Extended comparisons of best subset selection, forward stepwise selection, and the lasso. arXiv preprint arXiv:1707.08692, 2017.

[115] Trevor Hastie, Robert Tibshirani, and Martin Wainwright. Statistical learning with sparsity. Monographs on statistics and applied probability, 143:143, 2015.

[116] Hussein Hazimeh and Rahul Mazumder. Fast best subset selection: Coordinate descent and local combinatorial optimization algorithms. Operations Research, 68(5):1517–1537, 2020.

[117] Hussein Hazimeh and Rahul Mazumder. Learning hierarchical interactions at scale: A convex optimization approach. In International Conference on Artificial Intelligence and Statistics, pages 1833–1843. PMLR, 2020.

[118] Hussein Hazimeh, Rahul Mazumder, and Tim Nonet. L0learn: A scalable package for sparse learning using l0 regularization. arXiv preprint arXiv:2202.04820, 2022.

[119] Hussein Hazimeh, Rahul Mazumder, and Ali Saab. Sparse regression at scale: Branch-and-bound rooted in first-order optimization. arXiv preprint arXiv:2004.06152, 2020.

[120] Hussein Hazimeh, Rahul Mazumder, and Ali Saab. Sparse regression at scale: Branch-and-bound rooted in first-order optimization. Mathematical Programming, pages 1–42, 2021.

[121] Clifford Hildreth. Point estimates of ordinates of concave functions. Journal of the American Statistical Association, 49(267):598–619, 1954.

[122] Alan J Hoffman. On approximate solutions of systems of linear inequalities. Journal of Research of the National Bureau of Standards, 49(4):263, 1952.

[123] Mingyi Hong, Xiangfeng Wang, Meisam Razaviyayn, and Zhi-Quan Luo. Iteration complexity analysis of block coordinate descent methods. Mathematical Programming, 163(1-2):85–114, 2017.

[124] Peter J Huber. Robust estimation of a location parameter. The Annals of Mathematical Statistics, pages 73–101, 1964.

[125] Shibal Ibrahim, Wenyu Chen, Yada Zhu, Pin-Yu Chen, Yang Zhang, and Rahul Mazumder. Knowledge graph guided simultaneous forecasting and network learning for multivariate financial time series. In 2021 KDD Workshop on Machine Learning in Finance, 2021.

[126] Thorsten Joachims. Making large-scale svm learning practical. Technical report, Technical report, 1998.

[127] Andrew L Johnson and Daniel R Jiang. Shape constraints in economics and operations research. Statistical Science, 33(4):527–546, 2018.

[128] Franz Kappel and Alexei V Kuntsevich. An implementation of Shor's r-algorithm. Computational Optimization and Applications, 15(2):193–205, 2000.

[129] Hamed Karimi, Julie Nutini, and Mark Schmidt. Linear convergence of gradient and proximal-gradient methods under the polyak-łojasiewicz condition. In Joint European Conference on Machine Learning and Knowledge Discovery in Databases, pages 795–811. Springer, 2016.

[130] Heysem Kaya, Pinar Tüfekci, and Erdinç Uzun. Predicting CO and NO_x emissions from gas turbines: novel data and a benchmark pems. Turkish Journal of Electrical Engineering & Computer Sciences, 27(6):4783–4796, 2019.

[131] Heysem Kaya, Pmar Tüfekci, and Fikret S Gürgen. Local and global learning methods for predicting power of a combined gas & steam turbine. In Proceedings of the international conference on emerging trends in computer and electronics engineering icetcee, pages 13–18, 2012.

[132] Kshitij Khare, Sang-Yun Oh, and Bala Rajaratnam. A convex pseudolikelihood framework for high dimensional partial correlation estimation with convergence guarantees. Journal of the Royal Statistical Society: Series B: Statistical Methodology, pages 803–825, 2015.

[133] Arlene KH Kim, Adityanand Guntuboyina, and Richard J Samworth. Adaptation in log-concave density estimation. The Annals of Statistics, 46(5):2279–2306, 2018.

[134] Arlene KH Kim and Richard J Samworth. Global rates of convergence in log-concave density estimation. The Annals of Statistics, 44(6):2756–2779, 2016.

[135] Keiji Kimura and Hayato Waki. Minimization of akaike's information criterion in linear regression analysis via mixed integer nonlinear program. Optimization Methods and Software, 33(3):633–649, 2018.

[136] Achim Klenke. Probability Theory: A Comprehensive Course. Springer Science & Business Media, 2014.

[137] Roger Koenker and Ivan Mizera. Quasi-concave density estimation. The Annals of Statistics, 38(5):2998–3027, 2010.

[138] Gil Kur, Yuval Dagan, and Alexander Rakhlin. The log-concave maximum likelihood estimator is optimal in high dimensions. arXiv preprint arXiv:1903.05315v3, 2019.

[139] Gil Kur, Yuval Dagan, and Alexander Rakhlin. Optimality of maximum likelihood for log-concave density estimation and bounded convex regression. arXiv preprint arXiv:1903.05315, 2019.

[140] Gil Kur, Fuchang Gao, Adityanand Guntuboyina, and Bodhisattva Sen. Convex regression in multidimensions: Suboptimality of least squares estimators. arXiv preprint arXiv:2006.02044, 2020.

[141] Gil Kur and Eli Putterman. An efficient minimax optimal estimator for multivariate convex regression. In Conference on Learning Theory, pages 1510–1546. PMLR, 2022.

[142] M Paul Laiu and André L Tits. A constraint-reduced mpc algorithm for convex quadratic programming, with a modified active set identification scheme. Computational Optimization and Applications, 72(3):727–768, 2019.

[143] Hariharan Lakshmanan and Daniela Pucci De Farias. Decentralized resource allocation in dynamic networks of agents. SIAM Journal on Optimization, 19(2):911–940, 2008.

[144] Siu Kwan Lam, Antoine Pitrou, and Stanley Seibert. Numba: A llvm-based python jit compiler. In Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC, page 7. ACM, 2015.

[145] Jason Lee and Trevor Hastie. Structure learning of mixed graphical models. In Artificial Intelligence and Statistics, pages 388–396. PMLR, 2013.

[146] Jon Lee and Sven Leyffer. Mixed integer nonlinear programming, volume 154. Springer Science & Business Media, 2011.

[147] Wu Li. Error bounds for piecewise convex quadratic programs and applications. SIAM Journal on Control and Optimization, 33(5):1510–1529, 1995.

[148] Xuan Liang, Tao Zou, Bin Guo, Shuo Li, Haozhe Zhang, Shuyi Zhang, Hui Huang, and Song Xi Chen. Assessing beijing's pm2. 5 pollution: severity, weather impact, apec and winter heating. Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences, 471(2182):20150257, 2015.

[149] Eunji Lim and Peter W Glynn. Consistency of multidimensional convex regression. Operations Research, 60(1):196–208, 2012.

[150] Meixia Lin, Defeng Sun, and Kim-Chuan Toh. Efficient algorithms for multivariate shape-constrained convex regression problems. arXiv preprint arXiv:2002.11410, 2020.

[151] Jeff T Linderoth and Martin WP Savelsbergh. A computational study of search strategies for mixed integer programming. INFORMS Journal on Computing, 11(2):173–187, 1999.

[152] Dong C Liu and Jorge Nocedal. On the limited memory bfgs method for large scale optimization. Mathematical programming, 45(1):503–528, 1989.

[153] Yu Liu and Yong Wang. A fast algorithm for univariate log-concave density estimation. Australian & New Zealand Journal of Statistics, 60(2):258–275, 2018.

[154] Haihao Lu and Rahul Mazumder. Randomized gradient boosting machine. arXiv preprint arXiv:1810.10158, 2018.

[155] Zhi-Quan Luo and Paul Tseng. On the linear convergence of descent methods for convex essentially smooth minimization. SIAM Journal on Control and Optimization, 30(2):408–425, 1992.

[156] Zhi-Quan Luo and Paul Tseng. Error bounds and convergence analysis of feasible descent methods: a general approach. Annals of Operations Research, 46(1):157–178, 1993.

[157] Harry Markowitz. Portfolio selection. The Journal of Finance, 7(1):77–91, 1952.

[158] Rahul Mazumder, Arkopal Choudhury, Garud Iyengar, and Bodhisattva Sen. A computational framework for multivariate convex regression and its variants. Journal of the American Statistical Association, pages 1–14, 2018.

[159] Rahul Mazumder and Trevor Hastie. The graphical lasso: New insights and alternatives. Electronic journal of statistics, 6:2125, 2012.

[160] Rahul Mazumder, Peter Radchenko, and Antoine Dedieu. Subset selection with shrinkage: Sparse linear modeling when the snr is low. arXiv preprint arXiv:1708.03288, 2017.

[161] Rahul Mazumder, Peter Radchenko, and Antoine Dedieu. Subset selection with shrinkage: Sparse linear modeling when the snr is low. Operations Research, 2022.

[162] Peter McMullen. The maximum numbers of faces of a convex polytope. Mathematika, 17(2):179–184, 1970.

[163] Nicolai Meinshausen and Peter Bühlmann. High-dimensional graphs and variable selection with the Lasso. The Annals of Statistics, 34(3):1436 – 1462, 2006.

[164] Maethee Mekaroonreung and Andrew L Johnson. Estimating the shadow prices of so_2 and no_x for us coal power plants: a convex nonparametric least squares approach. Energy Economics, 34(3):723–732, 2012.

[165] Ramzi Ben Mhenni, Sébastien Bourguignon, Marcel Mongeau, Jordan Ninin, and Hervé Carfantan. Sparse branch and bound for exact optimization of l0-norm penalized least squares. In ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 5735–5739. IEEE, 2020.

[166] Sidhant Misra, Marc Vuffray, and Andrey Y Lokhov. Information theoretic optimal learning of gaussian graphical models. In Conference on Learning Theory, pages 2888–2909. PMLR, 2020.

[167] Ryuhei Miyashiro and Yuichi Takano. Subset selection by mallows' cp: A mixed integer programming approach. Expert Systems with Applications, 42(1):325–331, 2015.

[168] David R Morrison, Sheldon H Jacobson, Jason J Sauppe, and Edward C Sewell. Branch-and-bound algorithms: A survey of recent advances in searching, branching, and pruning. Discrete Optimization, 19:79–102, 2016.

[169] Balas Kausik Natarajan. Sparse approximate solutions to linear systems. SIAM journal on computing, 24(2):227–234, 1995.

[170] Ion Necoara, Yurii Nesterov, and Francois Glineur. Linear convergence of first order methods for non-strongly convex optimization. Mathematical Programming, pages 1–39, 2018.

[171] Arkadiĭ Semenovich Nemirovsky and David Borisovich Yudin. Problem Complexity and Method Efficiency in Optimization. Wiley, 1983.

[172] Yu Nesterov. Smooth minimization of non-smooth functions. Mathematical programming, 103(1):127–152, 2005.

[173] Yu Nesterov. Gradient methods for minimizing composite functions. Mathematical Programming, 140(1):125–161, 2013.

[174] Yurii Nesterov. Primal-dual subgradient methods for convex problems. Mathematical programming, 120(1):221–259, 2009.

[175] Yurii Nesterov. Lectures on Convex Optimization, volume 137. Springer, 2018.

[176] Jorge Nocedal and Stephen Wright. Numerical optimization. Springer Science & Business Media, 2006.

[177] Francisco Ortega and Laurence A Wolsey. A branch-and-cut algorithm for the single-commodity, uncapacitated, fixed-charge network flow problem. Networks: An International Journal, 41(3):143–158, 2003.

[178] Art B Owen. A robust hybrid of lasso and ridge regression. Contemporary Mathematics, 443(7):59–72, 2007.

[179] Brendan O'donoghue and Emmanuel Candes. Adaptive restart for accelerated gradient schemes. Foundations of computational mathematics, 15(3):715–732, 2015.

[180] Jayanta Kumar Pal, Michael Woodroofe, and Mary Meyer. Estimating a Polya frequency function. Lecture Notes-Monograph Series, pages 239–249, 2007.

[181] Ashwin Pananjady and Richard J Samworth. Isotonic regression with unknown permutations: Statistics, computation, and adaptation. arXiv preprint arXiv:2009.02609, 2020.

[182] Young Woong Park and Diego Klabjan. Subset selection for multiple linear regression via optimization. Journal of Global Optimization, 77(3):543–574, 2020.

[183] Jie Peng, Pei Wang, Nengfeng Zhou, and Ji Zhu. Partial correlation estimation by joint sparse regression models. Journal of the American Statistical Association, 104(486):735–746, 2009.

[184] Mert Pilanci, Martin J Wainwright, and Laurent El Ghaoui. Sparse learning via boolean relaxations. Mathematical Programming, 151(1):63–87, 2015.

[185] Boris T Polyak. Introduction to Optimization. Optimization Software Inc., Publications Division, New York, 1987.

[186] Fred Ramsey and Daniel Schafer. The statistical sleuth: a course in methods of data analysis. Cengage Learning, 2012.

[187] Garvesh Raskutti, Martin J Wainwright, and Bin Yu. Minimax rates of estimation for high-dimensional linear regression over ℓ_q-balls. IEEE transactions on information theory, 57(10):6976–6994, 2011.

[188] Fabian Rathke and Christoph Schnörr. Fast multivariate log-concave density estimation. Computational Statistics & Data Analysis, 140:41–58, 2019.

[189] Pradeep Ravikumar, Garvesh Raskutti, Martin J Wainwright, and Bin Yu. Model selection in gaussian graphical models: High-dimensional consistency of l1-regularized mle. In NIPS, pages 1329–1336, 2008.

[190] Albert Reuther, Jeremy Kepner, Chansup Byun, Siddharth Samsi, William Arcand, David Bestor, Bill Bergeron, Vijay Gadepally, Michael Houle, Matthew Hubbell, et al. Interactive supercomputing on 40,000 cores for machine learning and data analysis. In 2018 IEEE High Performance extreme Computing Conference (HPEC), pages 1–6. IEEE, 2018.

[191] Phillippe Rigollet and Jan-Christian Hütter. High dimensional statistics. Lecture notes for course 18S997, 2015.

[192] T. Robertson, F.T. Wright, and R. Dykstra. Order Restricted Statistical Inference. Probability and Statistics Series. Wiley, 1988.

[193] R Tyrrell Rockafellar. Convex Analysis. Princeton University Press, 1997.

[194] Adam J Rothman, Peter J Bickel, Elizaveta Levina, and Ji Zhu. Sparse permutation invariant covariance estimation. Electronic Journal of Statistics, 2:494–515, 2008.

[195] Richard J Samworth. Recent progress in log-concave density estimation. Statistical Science, 33(4):493–509, 2018.

[196] Richard J Samworth and Bodhisattva Sen. Editorial: Special Issue on "Nonparametric Inference under Shape Constraints". Statistical Science, 33:469–472, 2018.

[197] Richard J Samworth and Ming Yuan. Independent component analysis via nonparametric maximum likelihood estimation. The Annals of Statistics, 40(6):2973–3002, 2012.

[198] Sylvain Sardy, Andrew G Bruce, and Paul Tseng. Block coordinate relaxation methods for nonparametric wavelet denoising. Journal of computational and graphical statistics, 9(2):361–379, 2000.

[199] Dominic Schuhmacher, André Hüsler, and Lutz Dümbgen. Multivariate log-concave distributions as a nearly parametric model. Statistics & Risk Modeling with Applications in Finance and Insurance, 28(3):277–295, 2011.

[200] Emilio Seijo and Bodhisattva Sen. Nonparametric least squares estimation of a multivariate convex regression function. The Annals of Statistics, 39(3):1633–1657, 2011.

[201] Arseni Seregin and Jon A Wellner. Nonparametric estimation of multivariate convex-transformed densities. The Annals of Statistics, 38(6):3751–3781, 2010.

[202] Alexander Shapiro, Darinka Dentcheva, and Andrzej Ruszczyński. Lectures on stochastic programming: modeling and theory. SIAM, 2009.

[203] Naum Zuselevich Shor. Minimization Methods for Non-Differentiable Functions. Springer-Verlag, 1985.

[204] Noah Simon, Jerome Friedman, Trevor Hastie, and Rob Tibshirani. Regularization paths for cox's proportional hazards model via coordinate descent. Journal of statistical software, 39(5):1, 2011.

[205] Robert Tibshirani. Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society: Series B (Methodological), 58(1):267–288, 1996.

[206] Paul Tseng. Convergence of a block coordinate descent method for nondifferentiable minimization. Journal of optimization theory and applications, 109(3):475–494, 2001.

[207] Paul Tseng. On accelerated proximal gradient methods for convex-concave optimization. submitted to SIAM Journal on Optimization, 1, 2008.

[208] Pınar Tüfekci. Prediction of full load electrical power output of a base load operated combined cycle power plant using machine learning methods. International Journal of Electrical Power & Energy Systems, 60:126–140, 2014.

[209] Vladimir Vapnik. The nature of statistical learning theory. Springer science & business media, 1999.

[210] Hal R Varian. The nonparametric approach to demand analysis. Econometrica: Journal of the Econometric Society, pages 945–973, 1982.

[211] Roman Vershynin. High-dimensional probability: An introduction with applications in data science, volume 47. Cambridge university press, 2018.

[212] Martin J Wainwright. Information-theoretic limits on sparsity recovery in the high-dimensional and noisy setting. IEEE transactions on information theory, 55(12):5728–5741, 2009.

[213] Martin J Wainwright. High-dimensional statistics: A non-asymptotic viewpoint, volume 48. Cambridge University Press, 2019.

[214] Guenther Walther. Detecting the presence of mixing with multiscale maximum likelihood. Journal of the American Statistical Association, 97(458):508–513, 2002.

[215] Guenther Walther. Inference and modeling with log-concave distributions. Statistical Science, 24(3):319–327, 2009.

[216] Jie Wang, Jiayu Zhou, Peter Wonka, and Jieping Ye. Lasso screening rules via dual polytope projection. Advances in neural information processing systems, 26, 2013.

[217] Li Wang, Ji Zhu, and Hui Zou. Hybrid huberized support vector machines for microarray classification and gene selection. Bioinformatics, 24(3):412–419, 2008.

[218] Wei Wang, Martin J Wainwright, and Kannan Ramchandran. Information-theoretic bounds on model selection for gaussian markov random fields. In 2010 IEEE International Symposium on Information Theory, pages 1373–1377. IEEE, 2010.

[219] Yongqiao Wang and Shouyang Wang. Estimating α-frontier technical efficiency with shape-restricted kernel quantile regression. Neurocomputing, 101:243–251, 2013.

[220] Laurence A Wolsey and George L Nemhauser. Integer and combinatorial optimization, volume 55. John Wiley & Sons, 1999.

[221] Lin Xiao. Dual averaging methods for regularized stochastic learning and online optimization. Journal of Machine Learning Research, 11:2543–2596, 2010.

[222] Weijun Xie and Xinwei Deng. Scalable algorithms for the sparse ridge regression. SIAM Journal on Optimization, 30(4):3359–3386, 2020.

[223] Min Xu and Richard J Samworth. High-dimensional nonparametric density estimation via symmetry and shape constraints. The Annals of Statistics, 49:650–672, 2021.

[224] Fan Yang and Rina Foygel Barber. Contraction and uniform convergence of isotonic regression. Electronic Journal of Statistics, 13(1):646–677, 2019.

[225] Farzad Yousefian, Angelia Nedić, and Uday V Shanbhag. On stochastic gradient and subgradient methods with adaptive steplength sequences. Automatica, 48(1):56–67, 2012.

[226] Guo-Xun Yuan, Kai-Wei Chang, Cho-Jui Hsieh, and Chih-Jen Lin. A comparison of optimization methods and software for large-scale l1-regularized linear classification. The Journal of Machine Learning Research, 11:3183–3234, 2010.

[227] Cun-Hui Zhang. Risk bounds in isotonic regression. The Annals of Statistics, 30(2):528–555, 2002.

[228] Fuzhen Zhang. The Schur complement and its applications, volume 4. Springer Science & Business Media, 2006.

[229] Shuyi Zhang, Bin Guo, Anlan Dong, Jing He, Ziping Xu, and Song Xi Chen. Cautionary tales on air-quality improvement in beijing. Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences, 473(2205):20170457, 2017.

[230] Yuchen Zhang, Martin J Wainwright, and Michael I Jordan. Lower bounds on the performance of polynomial-time algorithms for sparse linear regression. In Conference on Learning Theory, pages 921–948. PMLR, 2014.