On the Power of Preconditioning in Sparse Linear Regression

Jonathan A. Kelner
MIT
Cambridge, USA
kelner@mit.edu

Frederic Koehler
Stanford University
Stanford, USA
fkoehler@stanford.edu

Raghu Meka
UCLA
Los Angeles, USA
raghum@cs.ucla.edu

Dhruv Rohatgi
MIT
Cambridge, USA
drohatgi@mit.edu

2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS) | 978-1-6654-2055-6/22/$31.00 ©2022 IEEE | DOI: 10.1109/FOCS52979.2021.00061

***Abstract*—Sparse linear regression is a fundamental problem in high-dimensional statistics, but strikingly little is known about how to efficiently solve it without restrictive conditions on the *design matrix*. We consider the (correlated) random design setting, where the covariates are independently drawn from a multivariate Gaussian $N(0, \Sigma)$, for some $n \times n$ positive semi-definite matrix Σ, and seek estimators $\hat{w}$ minimizing $(\hat{w} - w^*)^T \Sigma (\hat{w} - w^*)$, where w^* is the k-sparse ground truth. Information theoretically, one can achieve strong error bounds with only $O(k \log n)$ samples for arbitrary Σ and w^*; however, no efficient algorithms are known to match these guarantees even with $o(n)$ samples, without further assumptions on Σ or w^*.**

Yet there is little evidence for this gap in the random design setting: computational lower bounds are only known for worst-case design matrices. To date, random-design instances (i.e. specific covariance matrices Σ) have only been proven hard against the Lasso program and variants. More precisely, these "hard" instances can often be solved by Lasso after a simple change-of-basis (i.e. preconditioning).

In this work, we give both upper and lower bounds clarifying the power of preconditioning as a tool for solving sparse linear regression problems. On the one hand, we show that the preconditioned Lasso can solve a large class of sparse linear regression problems nearly optimally: it succeeds whenever the *dependency structure* of the covariates, in the sense of the Markov property, has low treewidth — even if Σ is highly ill-conditioned. This upper bound builds on ideas from the wavelet and signal processing literature. As a special case of this result, we give an algorithm for sparse linear regression with covariates from an autoregressive time series model, where we also show that the (usual) Lasso provably fails.

On the other hand, we construct (for the first time) random-design instances which are provably hard even for an optimally preconditioned Lasso. In fact, we complete our treewidth classification by proving that for *any* treewidth-t graph, there exists a Gaussian Markov Random Field on this graph such that the preconditioned Lasso, with any choice of preconditioner, requires $\Omega(t^{1/20})$ samples to recover $O(\log n)$-sparse signals when covariates are drawn from this model.

***Keywords*-sparse linear regression; preconditioning; high-dimensional statistics**

I. Introduction

In this paper, we study the fundamental statistical problem of *sparse linear regression with (correlated) random design*. In the simplest form of this problem, the learning algorithm is given access to m independent and identically distributed samples $(X_1, Y_1), \ldots, (X_m, Y_m)$ of the form

$$Y_i = \langle w^*, X_i \rangle + \xi_i \tag{1}$$

where each *covariate* $X_i \sim N(0, \Sigma)$ is a Gaussian random vector in $\mathbb{R}^n$, the noise $\xi_i \sim N(0, \sigma^2)$ is independent, and the true *coefficient vector* w^* is k-sparse, i.e. w^* has at most k nonzero entries. The goal of the learning algorithm is to output a vector w such that the *out-of-sample prediction error*

$$\mathbb{E}[(Y_0 - \langle w, X_0 \rangle)^2] = (w - w^*)^T \Sigma (w - w^*) + \sigma^2 \tag{2}$$

is as small as possible (i.e. close to σ^2, the error achieved by w^*), where (X_0, Y_0) is a fresh sample from the model.

There is a rich and vast body of work on sparse linear regression with ℓ_1-regularized approaches (see for example [59], [10], [5], [63]) ubiquitous in many domains and applied sciences (see e.g. [43], [55], [68], [24]). The problem is also extensively studied in the signal processing, compressed sensing and sketching communities (e.g. [22], [7], [2], [34], [12], [21], [11], [53]) where the measurements X are not necessarily Gaussian but may come from other structured distributions.

In the Gaussian random design setting, it is well known that *information theoretically*, it is possible to achieve error $(1 + \epsilon)\sigma^2$ with $m = O(k \log(n)/\epsilon)$ samples; note that the dependence on the ambient dimension n is logarithmic and that there is no dependence on the covariance matrix Σ. Unfortunately, despite a tremendous amount of work on sparse linear regression, we still do not know an efficient algorithm that for general (Σ, w^*) can get a small error (say $O(\sigma^2)$) even with up to $o(n)$ many samples. This limitation holds even when there is no noise (i.e., $\sigma = 0$).

The classical algorithmic results for this problem assume that the covariates satisfy some kind of well-conditioning property such as *incoherence* [22], or a variant such as the *Restricted Isometry Property* (RIP) [12], the *Restricted Eigenvalue Condition* [5], or the *Compatibility Condition* [63], and achieve up to constants the optimal statistical guarantee described above. See [63] for an extensive dis-

2575-8454/21/$31.00 ©2021 IEEE
DOI 10.1109/FOCS52979.2021.00061

cussion of these assumptions[1]. In particular, these conditions guarantee the success of ℓ_1-regularized methods for sparse linear regression such as the *Lasso* [59]: the Lasso estimator with tuning parameter λ is defined by the optimization problem

$$\operatorname*{argmin}_{w \in \mathbb{R}^n} \|Y - Xw\|_2^2 + \lambda \|w\|_1 \tag{3}$$

where $X : m \times n$ is the design matrix with rows X_i. The simplest to state version of these conditions, the RIP property, requires that all small submatrices of the covariance matrix are spectrally close to the identity matrix. When Σ is the identity matrix or has a bounded *condition number*[2], the restricted eigenvalue condition holds, and we can solve sparse linear regression with $O(k \log n)$ samples [51]. However, the above methods leave wide open what happens for general Σ. Since the population covariance Σ is given to us by nature in most statistical applications (for example, if the covariates X_i correspond to answers to survey questions, or observations from a complex scientific experiment), what happens for general Σ is a question of significant practical interest. This was one of the main motivations for studying weaker versions of the RIP property such as the Restricted Eigenvalue condition (see e.g. discussion in [5], [51], [36]) and compatibility condition [63], [62], and understanding how well the Lasso performs (well or not) with correlated design matrices remains an active area of research (see e.g. [16], [62], [39], [70], [3]).

While there are a few exceptions, such as settings where submodularity holds (e.g. [17], [18], [23]), we do not have good algorithms for dealing with ill-conditioned Σ. On the other hand, the state-of-the-art computational lower bounds for sparse linear regression [47], [69], [26], [33] apply only to the *fixed-design* setting with worst-case vectors X_i. It's unclear that extending these results to the random design setting is even possible, given various barriers to proving hardness of average case problems (see, for example, [1] [3]). Indeed, in the random design setting, the state-of-the-art lower bounds are simply against the Lasso [64], [27] or related classes of algorithms, such as linear regression with a coordinate-separable regularizer [70] or local search procedures[4] [29]. Such lower bounds by no means imply that the instances are computationally hard: even if the covariance matrix is ill-conditioned and Lasso fails, the sparse linear regression problem may still be tractable. Indeed, there are numerous examples [27], [16], [70], [38] of hard instances for Lasso which become solvable after a simple *change-of-basis*, and (to our knowledge) no examples of random designs which provably cannot be solved by Lasso after such a change-of-basis.

[1]In the fixed design setting, these conditions are placed on the empirical covariance matrix (or equivalently, the design matrix). The results of [51], [71] shows the analogous conditions on the population covariance are inherited by the empirical covariance matrix in the random design setting.

[2]The ratio of the largest eigenvalue to the smallest eigenvalue.

[3]This paper discusses obstacles to improper learning; however, in random-design sparse linear regression where Σ is known, an improper learning algorithm can be converted into a proper learning algorithm (by using the former to generate artificial samples, and then running (ordinary) linear regression).

[4]We note that this last work is focused on understanding a constant factor gap in the isotropic setting, a related but fairly different goal vs. understanding the landscape for general Σ.

A. Preconditioned Lasso

Preconditioning is a powerful and extremely well-studied technique for solving linear systems. In that literature, there are two types of preconditioning: from the left, or from the right [54].

In the vast literature on ℓ_1 methods for sparse linear regression, we are aware of no works that systematically study the power of "right" preconditioning, i.e. an initial change-of-basis in parameter space (see Section III for further discussion, including the substantial differences between "left" and "right" preconditioning in sparse linear regression). Are there natural classes of sparse linear regression problems which can be solved by a preconditioned ℓ_1 method but not by classical methods? Are there examples of designs which provably cannot be helped by appropriate preconditioning? In this paper, we systematically study these questions. To formalize the notion of an initial change-of-basis, we define the following large and natural class of convex programs, which we call the *preconditioned Lasso*.

Definition I.1 (Preconditioned Lasso)**.** Let $S \in \mathbb{R}^{n \times s}$ be a matrix. The *S-preconditioned Lasso* on samples $(X_i, Y_i)_{i=1}^m$ with tuning parameter λ is the program

$$\operatorname*{argmin}_{w \in \mathbb{R}^n} \|Y - Xw\|_2^2 + \lambda \left\|S^T w\right\|_1 \tag{4}$$

where $X : m \times n$ is the design matrix with rows X_i. Taking $\lambda \to 0$, as is done for noiseless samples, yields the *S-preconditioned Basis Pursuit (BP)*:

$$\operatorname*{argmin}_{w \in \mathbb{R}^n : Xw = Y} \left\|S^T w\right\|_1. \tag{5}$$

Programs 4 and 5 are convex, so can be solved in time $\mathrm{poly}(n, s, m)$. If S is the identity matrix, then they are just the well-studied Lasso (see (3)) and Basis Pursuit programs.

Program 4 has been previously studied in the literature, under various names including the *generalized Lasso* [60]. However, in most applications of the generalized Lasso the motivation is different: the matrix S is introduced into the program because the signal is not sparse in the original basis, but in a different one (e.g. for piecewise constant signals, S is chosen to give the *total variation* norm which penalizes the discrete derivative [60], [48]). In contrast, we are only interested in recovering signals *sparse in the original basis*, and we seek to choose S based on the design matrix to improve the performance of the Lasso. To avoid confusion, we therefore refer to this program as the "preconditioned Lasso" in this paper. This should not be confused with a

different and largely unrelated terminology introduced in prior work [67], [37]; we expand on this distinction in Section III.

As the name suggests, the above class of programs essentially corresponds to solving the Lasso after first performing an appropriate change of basis. Indeed, if S is an $n \times n$ invertible matrix, then the S-preconditioned Lasso is equivalent to Lasso with a (right) preconditioned design:

$$\operatorname*{argmin}_{u \in \mathbb{R}^n} \|Y - X(S^T)^{-1}u\|_2^2 + \lambda \|u\|_1.$$

This is a natural class since, as previously remarked, the ability to change basis is powerful enough to fix the Lasso in several examples where it is otherwise known to fail (for instance, see examples in [27], [16], [70], [38]).

As we will explain further, in this paper we present both upper and lower bounds for this class of programs. Our results are closely tied to a standard notion of *graphical structure* for the covariate distribution. Before explaining the conditions in general, we start with a motivating example: estimating a sparse linear functional of a simple random walk, studied in [39].

B. A motivating example: Random walk/Brownian motion

Suppose we have a sequence of random variables $R_1, \ldots, R_n$ where each R_i is generated from R_{i-1} and some independent noise. That is, $Z_1, \ldots, Z_n \sim N(0,1)$ are independent Gaussian random variables, with $R_1 = Z_1$ and

$$R_i = R_{i-1} + Z_i$$

for $i > 1$. This describes a simple random walk, one of the simplest forms of time-series data. If each covariate vector X_i is an i.i.d. copy of $(R_1, \ldots, R_n)$, then the covariance matrix Σ is just $\Sigma_{ij} = \min(i,j)$. More importantly, Σ is quite ill-conditioned and existing guarantees (e.g., restricted eigenvalue etc.) do not seem useful in this scenario [39]. In the work [39], the authors gave upper bounds on the performance of the Lasso for this version of sparse linear regression, which did not match the performance of the information-theoretically optimal algorithm.

One of the technical innovations in the present paper is a general and relatively easy-to-use method for proving *lower bounds* on the performance of the Lasso in random design problems. As a simple application of our general result, we clarify the behavior of the Lasso in this model by proving a strong negative result:

Theorem I.2. *For any $k \geq 2$, there is a k-sparse signal $w^* \in \mathbb{R}^n$ such that the Lasso and Basis Pursuit require at least $m = \Omega(\sqrt{n})$ samples to exactly recovery w^* from noiseless observations $(X_i, Y_i)_{i=1}^m$, when the covariates X_i are independently drawn from the Gaussian random walk $N(0,\Sigma)$ and $Y_i = \langle w^*, X_i \rangle$. The same holds if the coordinates of the covariates are normalized to all have variance* 1.

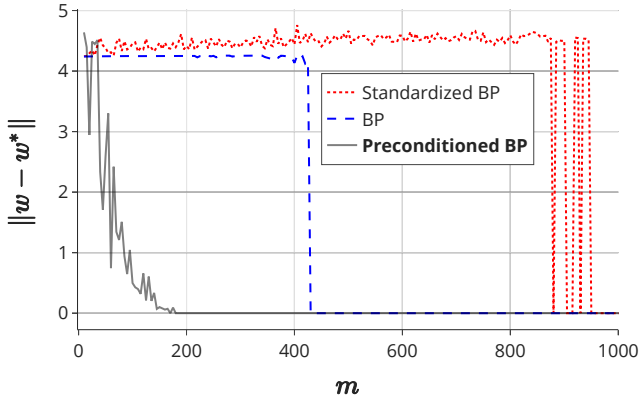

Figure 1: Basis Pursuit (BP) vs Preconditioned BP vs Standardized BP in the case of the simple random walk with $n = 32768$ (2^{15}) variables. The x-axis is the number of samples m (incremented in steps of 5) and the y-axis is the error in recovering w^* in Euclidean norm in a single independent run of the algorithms. Consistent with our theory (Theorem I.2 and Theorem I.5), preconditioned BP succeeds at exact recovery with significantly fewer samples than normal BP or standardized BP (BP with the coordinates of X_i standardized to variance 1, which is a common preprocessing step). The ground truth covariates X_i are i.i.d. copies of a simple random walk with Gaussian steps, and the ground truth labels are $Y_i = 3[(X_i)_{32768} - (X_i)_{32767}]$, i.e. $w^* = (0, \ldots, 0, -3, 3)$.

The above theorem shows that the most popular algorithmic approach for sparse linear regression problems, ℓ_1-regularized least squares, performs poorly in this problem; in fact, its sample complexity is exponentially sub-optimal in the ambient dimension n (this was also observed experimentally in [38]). This brings up an obvious question, which to the best of our knowledge, was unanswered even in this particular case — can *any* polynomial time algorithm achieve nearly optimal performance (or even $o(\sqrt{n})$ sample complexity) in this example?

We show the answer is *yes* — sparse linear regression on a random walk can be solved efficiently. While Lasso run in the usual way fails, it turns out that if we first make an appropriate change-of-basis, i.e. precondition, Lasso will succeed. More formally, there is a sparse "preconditioner" matrix $S \in \mathbb{R}^{n \times n}$, such that the S-preconditioned Lasso recovers any k-sparse rule $w^* \in \mathbb{R}^n$ from $m = O(k \log^3 n)$ samples $Y_i = \langle w^*, X_i \rangle + \xi_i$, where ξ_i is independent Gaussian noise. In the present case we actually choose S to be an invertible matrix, so this can be interpreted simply as a change of basis.

Theorem I.3 (Special case of Theorem I.5)**.** *Suppose that $X_1, \ldots, X_m$ are independent copies of $(R_1, \ldots, R_n)$ and $Y_i = \langle w^*, X_i \rangle + \xi_i$ with $\xi_i \sim N(0, \sigma^2)$ independent and $\sigma^2 \geq 0$. There is a polynomial-time algorithm (preconditioned Lasso/BP) which outputs $\hat{w}$ such that with high probability,*

$$(\hat{w} - w^*)^T \Sigma (\hat{w} - w^*) = O\left(\frac{\sigma^2 k \log^2(n)}{m}\right) \qquad (6)$$

provided $m = \Omega(k \log^3(n))$. (When $\sigma = 0$, the rhs is zero.)

The preconditioner S^T which works is quite simple, and builds upon inspiration from other settings in the compressed sensing and signal processing literature (see [45], [48], [35]). The inverse $(S^T)^{-1}$ is the composition of the differencing operator $D[x]_i = x_i - x_{i-1}$ with the *discrete Haar wavelet transform* [32], [45]. The first transformation (differencing) transforms the sequence $R_1, \ldots, R_n$ back into the steps $Z_1, \ldots, Z_n$, which completely fixes the ill-conditioning of the basis. However, the first step destroys sparsity: $\langle w^*, R \rangle$ is not a sparse linear functional of Z, but instead a dense linear functional with *piecewise constant* coefficients. The second step, the Haar transform, is orthogonal — hence preserves well-conditioning — and it restores sparsity, because a piecewise constant signal is sparse in the Haar basis [45]. Given this, Theorem I.3 follows from classical results on sparse linear regression with isotropic covariates from the compressed sensing literature (e.g. [10], [5]).

It is remarkable that the random walk model admits a preconditioning matrix S which combines so well with the Lasso. This suggests the question:

Question I.4. *For which Σ can we construct a preconditioner such that the preconditioned Lasso solves sparse linear regression for covariates drawn from $N(0, \Sigma)$?*

C. Preconditioning and Dependency Graphs

The answer, as it turns out, depends on the *conditional independence structure* of $X \sim N(0, \Sigma)$, or equivalently the *dependency graph* of the distribution. We first introduce this notion.

Fix a distribution D on $\mathbb{R}^n$, and a graph G on n vertices. We say D satisfies the *Markov property* with respect to G if the following holds for $X \sim D$: whenever i, j are not adjacent in G, X_i, X_j are independent conditioned on $(X_k : k \text{ a neighbor of } i \text{ in } G)$. That is, G is a *dependency graph* for the distribution D. The study of dependency graphs of distributions has a rich history and vast literature within statistics and machine learning, under the general area of *graphical models* — see e.g. [41], [19], [6], [66], [49].

For a multivariate Gaussian distribution $N(0, \Sigma)$ with invertible Σ, there is a clean characterization of the dependency graph. Let $\Theta = \Sigma^{-1}$ be its *precision matrix*. The dependency graph of $N(0, \Sigma)$ is precisely the graph whose adjacency matrix is the support of Θ [41], [19]. Reconsidering the example of random walks, a key property of the distribution of $(R_1, \ldots, R_n)$ as defined above is the following *Markov property*: conditional on R_i, the past variables $R_1, \ldots, R_{i-1}$ are independent of the future variables $R_{i+1}, \ldots, R_n$. Equivalently (see e.g., [41]), the precision matrix $\Theta = \Sigma^{-1}$ of $R_1, \ldots, R_n$ is supported on the adjacency matrix of the path graph.

Our main contribution is an essentially complete answer to Question I.4 in terms of the corresponding dependency graph of Σ. At a high level, we show that whenever the dependency graph of Σ has small *treewidth*, then there is a preconditioner such that Lasso succeeds. Conversely, we show that for any graph G with high treewidth, there is a Gaussian distribution with G as the dependency graph on which *no* preconditioning can make Lasso succeed. This shows that treewidth, long used as a natural complexity measure in graphical models, e.g. in the context of the celebrated junction tree algorithm [42], also determines the difficulty of solving a sparse linear regression with preconditioned Lasso. We formally state our results next.

D. Main Results

Preconditioning for small treewidth: We show that whenever the dependency graph of the covariate distribution has low treewidth, say t, there exists a choice of preconditioner which makes the Lasso succeed with $\gg kt \log^3 n$ samples. Furthermore, such a preconditioner can be constructed efficiently without exact knowledge of Σ: just knowing the dependency graph allows us to efficiently construct the preconditioner based off of the samples. Formally, we show the following:

Theorem I.5. *Let G be a graph on $[n]$, and let $\Theta \in \mathbb{R}^{n \times n}$ be a positive-definite matrix supported on G. Suppose that G has treewidth at most t. Let $\sigma \geq 0$. Then there is a polynomial-time algorithm which outputs $\hat{w}$ such that with high probability,*

$$(\hat{w} - w^*)^T \Sigma (\hat{w} - w^*) = O\left(\frac{\sigma^2 kt \log^{1/2}(t) \log^2(n)}{m}\right) \qquad (7)$$

from (1) knowledge of the graph G, and (2) $m = \Omega(kt \log^3 n)$ independent samples (X_i, Y_i), where $Y_i = \langle w^*, X_i \rangle + \xi_i$ with w^* a k-sparse vector, and independently $X_i \sim N(0, \Theta^{-1})$ and $\xi_i \sim N(0, \sigma^2)$.

Remark 1. The extra $\log^{1/2}(t)$ factor arises from the approximation algorithm of [25] and can be eliminated if the optimal tree decomposition is given as input. Also, in the full version of the paper we show how to shave the extra $\log(n)$ factor from (7) by combining our preconditioner with model-based Iterative Hard Thresholding instead of the Lasso (cf. [2]). Finally, we note the results generalize straightforwardly to subgaussian data and noise, in which case the sparsity pattern of Θ may differ from the graphical structure according to the Markov property.

Besides giving a characterization of dependency structures which enable the success of preconditioned Lasso, the above also covers several important cases that arise in practice. The simplest case is the random walk discussed above, where the dependency graph is a path and therefore has treewidth 1. Especially if we are regressing on time series data, the path graph may sometimes be a reasonable assumption on the dependency structure of the covariates. However, even in the specific context of time series, one often has multiple interacting time series and/or longer range interactions (consider e.g. an $AR(2)$ model [9]) which fundamentally change the graph structure. In these situations, the treewidth is bounded by the length of the interactions, and thus may naturally be small. More generally, sparse graphical structure is often a natural assumption in practice and plays, for example, a very important role in causal inference and reasoning [49], [50].

Failure of preconditioning for high treewidth: We complement our upper bound with a sample complexity lower bound for high-treewidth graphs: for *any* graph G with treewidth t, there is a multivariate Gaussian distribution with dependency graph G such that *for any preconditioner* S, the S-preconditioned Lasso fails (with high probability) unless the number of samples is $\Omega(t^c)$ for an absolute constant $c > 0$. The preconditioner is allowed to depend on the distribution, and the lower bound result holds in the (easiest) noiseless setting, where the corresponding notion of success requires exact recovery of the ground truth.

Theorem I.6. *Pick $n, t, s \in \mathbb{N}$, and suppose that G is a graph on $[n]$ with treewidth at least t. Then there exists $k = O(\log n)$ and some positive-definite precision matrix Θ, supported on G, with condition number $\mathrm{poly}(n)$, such that the following holds: for every preconditioner $S \in \mathbb{R}^{n \times s}$, the S-preconditioned basis pursuit requires $m = \Omega(t^{1/20})$ samples (X_i, Y_i) to exactly recover a k-sparse coefficient vector w^* from covariates $X_1, \ldots, X_m$ drawn i.i.d. from $N(0, \Theta^{-1})$ and noiseless responses $Y_i = \langle w^*, X_i \rangle$, with probability better than $1/t^{1/400}$.*

To the best of our knowledge, this result provides the first class of examples of random design problems where a change of basis *provably cannot* fix the performance of the Lasso. To prove this result, we develop an easy-to-use machinery for proving lower bounds on the performance of the Lasso in random design settings, which is of independent interest.

II. Overview of Techniques

A. Algorithms for Low-treewidth

Known Σ setting: First, we describe the simplified version of the low-treewidth algorithm, which assumes knowledge of the population covariance matrix Σ. The algorithm generalizes the one for the path described earlier. We show that there exists a sparse matrix S such that $\Sigma = SS^T$ (i.e., a *sparse Cholesky factorization*) and such that both S^T and $(S^T)^{-1}$ are *sparsity preserving* (within $poly(t, \log(n))$ factors, where t is the treewidth). This enables us to precondition the Lasso exactly, transforming from the original problem to a new problem where the covariates have identity covariance, and the unknown signal is transformed but still sparse — the ideal setting to apply classical results on sparse linear regression from compressed sensing literature.

The construction of the preconditioner S is via a version of the *nested dissection* method, originally designed for quickly solving systems of linear equations over low-treewidth graphs [30], [44], although the actual analysis we need to perform is fairly different from those works. This kind of recursive decomposition is morally related to the Haar wavelet transform and its generalization to trees (see e.g. [56]) and variants which apply to low-treewidth graphs, e.g. [20] though the details and motivation differ.

Concretely, the tree we start with is given by computing a *tree decomposition* of our low-treewidth graph, using for example the approximation algorithm of [25]. This tree provides a natural hierarchical decomposition of the graph, because we can always break a tree into roughly equal size pieces by removing its *centroid* [31]. We can then exploit the Markov property to reduce the problem of preconditioning the entire model to preconditioning each of the smaller pieces. Recursing, we get a sparse block Cholesky factorization of Σ that we use as the preconditioner.

Because our preconditioner has a natural tree structure, we also show that we can use algorithmic tools from the area of *model-based compressed sensing* (see, e.g., [2]) to shave an extra log factor from the rate that arises when using the preconditioned Lasso. This algorithm, based on a version of Iterative Hard Thresholding [7], allows us to recover the information-theoretically optimal $O(\sigma^2 k \log(n)/m)$ rate in the bounded treewidth setting.

Data-Dependent Sparse Preconditioner: It's often the case that the true population covariance matrix Σ is unknown to the algorithm. Furthermore, since we assume access to only a small number of samples X_i from the covariate distribution, the empirical covariance matrix cannot stand in as a suitable replacement for the true matrix Σ (for example, the empirical covariance matrix may not be invertible even if the true Σ is).

We show how to overcome these difficulties under the more realistic assumption that the graphical structure of the distribution, i.e., the support of the precision matrix $\Theta = \Sigma^{-1}$, is known. This kind of modeling assumption is prevalent in the causal inference and graphical models literature (see, e.g., [49]) and is generally more plausible than knowing the exact matrix Σ. Also, even if initially unknown, the graph structure may be recoverable from a small number of samples using GGM learning algorithms (e.g., [46], [28], [38]).

Algorithmically, we build our preconditioner by performing an approximate *block Cholesky factorization* of the empirical covariance matrix $\hat{\Sigma}$, following the tree structure described above. By using the Markov property, we can handle the poor approximation quality of the empirical covariance matrix $\hat{\Sigma}$ to Σ by zeroing out all of the entries of various Schur complements which arise during the Cholesky factorization and which must be zero due to the Markov property. If A is the centroid from the tree decomposition and P, Q are the resulting subforests given by removing A, the approximate block factorization is given by

$$S := \begin{bmatrix} \tilde{\Sigma}_{AA}^{1/2} & 0 & 0 \\ \tilde{\Sigma}_{PA}\tilde{\Sigma}_{AA}^{-1/2} & S_P & 0 \\ \tilde{\Sigma}_{QA}\tilde{\Sigma}_{AA}^{-1/2} & 0 & S_Q. \end{bmatrix}$$

where $\tilde{\Sigma}$ is the empirical covariance matrix, and the bottom right is a (recursively defined) approximate block Cholesky factorization of the Schur complement $\tilde{\Sigma}/A$ with zerod out bottom-left and top-right sub-blocks. This factorization is *not* a spectral approximation of the empirical covariance matrix $\tilde{\Sigma}$ (which may be rank degenerate); instead, we show that changing basis by S results in a new Lasso problem which satisfies the *Restricted Isometry Property* [12]. The proof of this fact is quite involved, as we need to precisely track the accumulation of errors in the factorization from the perspective of a sparse test vector.

Aside: sparse linear regression with sparse covariance: The assumption that Θ is sparse is very common and natural from a modeling perspective. That being said, we also consider what happens when Σ, instead of Θ, is sparse. In the full version of the paper, we give an algorithm for k-sparse linear regression with runtime roughly $d^k \cdot poly(n)$, where the rows of Σ are d-sparse. It's again based on preconditioning the Lasso but uses a randomized preconditioner based on a *site percolation* process on the graph (see, e.g., [40]), where each vertex of the graph is kept with probability p.

B. Impossibility of preconditioning in high-treewidth models

In this section, we outline the proof of Theorem I.6, the sample complexity lower bound for high-treewidth graphs. There are three main elements to the proof:

1) Identifying conditions on a precision matrix $\Theta = \Sigma^{-1}$ and preconditioner S, under which the S-preconditioned Lasso will fail (for some sparse signal, with covariates drawn from $N(0, \Sigma)$)
2) Constructing a precision matrix on (a slight variant of) the *grid graph* which satisfies these conditions for any preconditioner
3) Extending the lower bound for the grid graph variant, in a black-box manner, to a lower bound for any high-treewidth graph

Conditions under which (preconditioned) Lasso fails: There are two distinct reasons why classical Lasso might fail to recover some signal: either the covariates are ill-conditioned, or the ground truth is not sparse. For preconditioned Lasso, the situation is *roughly* analogous, and we have two cases: if the preconditioned covariates are ill-conditioned, then recovery should intuitively fail; and if the preconditioner has dense rows, meaning that the ground truth may be dense in the preconditioned basis, then recovery should intuitively fail. While making these statements precise requires additional assumptions, this intuition is accurate in spirit.

We first formalize the first case above. For a fixed design matrix X, the standard KKT conditions can determine whether Lasso/Basis Pursuit succeed at exact recovery (see, e.g., Theorem 7.8 of [65]). However, to show the Lasso fails in random design, we need a condition on Σ that guarantees failure with a high probability over X. Despite a vast literature on conditions for success and failure of Lasso, we are not aware of a broad, sufficient condition on the covariance matrix in the random design setting under which there must exist some sparse signal that causes Lasso to fail (even in the more straightforward non-preconditioned setting). We rectify this gap by introducing the *Weak (S-Preconditioned) Compatibility Condition*. This condition is defined analogously to stronger compatibility conditions (cf. [51], [63]), which are sufficient for Lasso's success. That is, it roughly states that SS^T (identity in the case of unpreconditioned Lasso) approximates Σ. However, unlike classical compatibility conditions, the condition we introduce is *necessary*[5] as opposed to sufficient: if it is not satisfied, then the S-preconditioned Lasso will fail with high probability on some sparse signal.

To be concrete, we describe the Weak Compatibility Condition and why it is necessary for the success of Lasso. When $S = I$ and given m samples, the Weak Compatibility Condition is said to fail when there is a sparse vector w^* and an $\Omega(m)$-dimensional subspace U such that for all $u \in U \setminus \{0\}$, the quantity $(u^T \Sigma u)/\|u\|_1^2$ is much larger than $((w^*)^T \Sigma w^*)/\|w^*\|_1^2$. Informally, this means that it is much cheaper in ℓ_1 norm to use features from the subspace U than from the direction w^*. We show that if the Weak Compatibility Condition fails to hold and w^* is the ground-truth signal, the Lasso will fail with high probability to recover w^*. This is because we can overfit the signal by finding a $u \in U$ such that $Xu = Xw^*$, and we can show that there will exist at least one such u with smaller ℓ_1 norm than the ground truth w^*. The S-Preconditioned version of the

[5]An interesting result with related motivation is Theorem 3.1 of [3], which shows that if the Lasso succeeds for arbitrary sparse signals while (a variant of) the ℓ_1-eigenvalue/compatibility constant of the design matrix is large, then the regularization parameter λ must be small. However, it leaves open the possibility that Lasso may succeed with an appropriately small choice of λ.

condition replaces the ℓ_1 norm by the general $u \mapsto \|S^T u\|_1$ norm, and formalizes the necessary condition for SS^T to approximate Σ well.

Of course, it's easy to construct a preconditioner S such that SS^T does approximate (or even equals) Σ; it just might not be sparsity-preserving (i.e., $S^T w^*$ need not be sparse even for sparse w^*). We formalize the intuition that a preconditioner with dense rows should also cause preconditioned Lasso to fail (i.e., the second mode of failure we alluded to above). This is surprisingly challenging. The main obstacle is that this is false without additional restrictions: for example, replacing the preconditioner S with the column-wise concatenation $[S; S]$ doubles the size of the support of $S^T w^*$ but doesn't affect the output of the S-preconditioned basis pursuit, so sample complexity cannot be directly tied to $|\operatorname{supp}(S^T w^*)|$.

More technically, even if $S^T w^*$ is dense, we cannot simply change the basis and use the fact that classical Lasso fails on dense signals[6]. The issue is that S^T may map to a higher-dimensional space, so changing basis introduces a new subspace constraint into the program, which could make KKT optimality conditions easier to satisfy. Instead, to find violations of the KKT optimality conditions, we must use additional structural properties of Θ and S (e.g., that every *column* of S is either dense or has a very small norm). In the next paragraph, we discuss a specific framework for constructing Θ; it is under this framework that we can show that a dense preconditioner causes recovery to fail.

Framework for constructing Θ: How do we construct a positive-definite matrix Θ such that any preconditioner S is either dense or poorly approximates Θ (i.e. SS^T is spectrally far from $\Sigma = \Theta^{-1}$)? Obviously, Θ must be ill-conditioned. Taking this intuition to the extreme, we can consider a nearly-degenerate matrix $\Theta = \tilde{\Theta} + \epsilon I$, where $\tilde{\Theta}$ is PSD with an r-dimensional kernel, and ϵ is arbitrarily small. If the preconditioner S satisfies $SS^T \approx \Sigma$ in an appropriate sense, then it can be seen that every column of S lies arbitrarily near $\ker \tilde{\Theta}$ (as $\epsilon \to 0$) and that the columns must, in the limit, span $\ker \Theta$. So for S to necessarily be dense, it's enough that $\ker \tilde{\Theta}$ is high-dimensional and contains no sparse vectors. This is the key insight in understanding what properties a hard instance should have:

To construct a precision matrix $\Theta = \tilde{\Theta} + \epsilon I$ which is hard to precondition, it suffices to show that $\ker \tilde{\Theta}$ is high-dimensional and dense, i.e., contains no sparse vectors.[7]

[6]Recall that S-preconditioned Lasso can be intuitively viewed as standard Lasso where the actual signal is $S^T w^*$.

[7]In a way, this property of $\tilde{\Theta}$ resembles the Restricted Isometry Property, which is a property of the covariance matrix Σ that enables the *success* of Lasso. However, in our case, the condition is placed on the inverse covariance matrix Θ, which means it obstructs preconditioning the large eigendirections of Σ.

Of course, this "story" has several issues. First, the assumption that SS^T spectrally approximates Σ is too strong because the converse does not imply that S-Preconditioned Lasso fails. Instead, we only assume that the S-Preconditioned Weak Compatibility Condition holds, which introduces new difficulties in proving that S is dense. Notably, a vital step of the proof requires that Θ is *very sparse*. This further motivates our investigation of sparse linear regression when covariates are drawn from Gaussian Graphical Models: the sparse dependency structure is crucial.

Second, we do not want Θ to be arbitrarily close to degenerate; we want it to have $\operatorname{poly}(n)$ condition number. This introduces a new wrinkle, but the same insight still mostly holds: we want to find a PSD matrix $\tilde{\Theta}$ (supported on some graph), such that the kernel is high-dimensional and is "robustly" dense, i.e., not too close to any sparse vector. This matrix should also have a polynomial condition number on $\operatorname{span}(\tilde{\Theta})$. The following theorem formalizes the above framework, i.e., conditions on Θ under which S-preconditioned Lasso cannot succeed:

Theorem II.1. *Let $\tilde{\Theta} \in \mathbb{R}^{n \times n}$ be a PSD matrix. Let $k, m, s > 0$. Let $\tau > 0$ and $V \subseteq [n]$ and let η be the infimum of $\|x_V - y\|_2 / \|x\|_2$ over all nonzero $x \in \ker(\tilde{\Theta})$ and τ-sparse $y \in \mathbb{R}^V$. Also, let λ be the smallest non-zero eigenvalue of $\tilde{\Theta}$. Suppose that the following hold:*

- *The rows (and columns) of $\tilde{\Theta}$ are k-sparse*
- *$r := \dim \ker(\tilde{\Theta}) > 2m$*
- *$k > 3(|V|/\tau)\log(n)$*

Pick any positive $\epsilon < \eta^2 \lambda^3 / (16200 n^3 \|\tilde{\Theta}\|_F^2)$. Define $\Theta = \tilde{\Theta} + \epsilon I$. For any preconditioner $S \in \mathbb{R}^{n \times s}$, there is some k-sparse signal such that S-preconditioned Lasso fails at exact recovery with probability at least $1 - \frac{4m}{3r} - \exp(-\Omega(m))$, from independent covariates $X_1, \ldots, X_m \sim N(0, \Theta^{-1})$ and noiseless responses $Y_i = \langle w^*, X_i \rangle$.

The expander graph: Ultimately, we will need to construct a positive semi-definite matrix $\tilde{\Theta}$ supported on a variant of the grid graph, whose kernel has the above properties. However, we first discuss a simple construction on an expander graph, which shares several ideas with the more intricate grid graph construction. Our approach is to define $\tilde{\Theta} = M^T M$ for an appropriate matrix $M \in \mathbb{R}^{n-r \times n}$. Then $\tilde{\Theta}$ is necessarily PSD, and its kernel must have dimension at least r. As $\ker(\tilde{\Theta}) = \ker(M)$, we can view each row of M as an equation that constrains the kernel. Specifically, we let each row of M be a sparse Bernoulli random vector. With high probability, M is the adjacency matrix of a bipartite expander graph, and classical results show that $\ker(M)$ is robustly dense.

This construction is noteworthy in several ways. First, unlike the lower bounds achieved in Theorem I.6, this construction yields a linear sample complexity lower bound (in

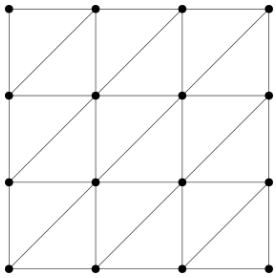

Figure 2: 4×4 simplicized grid graph

the number of variables n, or equivalently in the treewidth) for recovering polylog(n)-sparse vectors, for the expander graph. Second, the proof utilizes well-known results from compressed sensing and random matrix theory, providing an interesting example of how techniques originally developed to prove the *success* of sparse recovery methods can be applied to establish *failure*. Third, this construction may be a good candidate for stronger lower bounds (e.g., computational or statistical query).

The grid graph: We now turn to constructing a positive-semidefinite matrix $\tilde{\Theta}$ supported on the grid graph, such that the kernel has the above properties. As with the expander graph, we define $\tilde{\Theta} = M^T M$ for an appropriate matrix $M \in \mathbb{R}^{n-r \times n}$.

The requirement that $\tilde{\Theta}$ be supported on the grid graph means that the equations must be in some sense "local". An entry $\tilde{\Theta}_{ij}$ is nonzero if there is some equation containing both variables i and j. This is problematic if we want $\tilde{\Theta}$ to truly be supported on the grid graph (since it is triangle-free, no equation could have more than 2 variables). Instead, we relax that condition slightly, to require that $\tilde{\Theta}$ be supported on the *simplicized grid graph*, depicted in Figure 2 for $n = 4$.

Now, for every triangle of the simplicized grid graph, there can be an equation constraining the triangle's vertices. We define a subset $\mathcal{X}$ of the top row and a subset $\mathcal{Y}$ of the bottom row. We construct equations so that $\mathcal{X}$ is a set of free variables, and every other vertex has precisely one constraint, and any solution is robustly dense on either $\mathcal{X}$ or $\mathcal{Y}$, which suffices for our needs.

That such a construction exists is a priori unclear; for example, if the equations have random weights, then it turns out that in most solutions, the norm on row r decays exponentially as r increases. Hence, if the variables in $\mathcal{X}$ are set to some sparse vector, then $\mathcal{Y}$ will be very close to 0 and thus not robustly dense. Similar issues arise if the equations' weights are periodic; e.g., if the first row avoids high-frequency Fourier vectors, then subsequent rows may decay exponentially.

Instead of these approaches, we take inspiration from a simple construction that does not observe the "locality" conditions but instead is essentially a complete bipartite graph between $\mathcal{X}$ and $\mathcal{Y}$. Specifically, if there are no conditions on the locality of the equations, then we may introduce constraints so that each variable in $\mathcal{Y}$ is a Gaussian random linear combination of the variables in $\mathcal{X}$, i.e., $v_{\mathcal{Y}} = A v_{\mathcal{X}}$ where A is a Gaussian random matrix, and v is any solution. Standard matrix concentration results imply that if v is nonzero, then either $v_{\mathcal{Y}}$ or $v_{\mathcal{X}}$ must be robustly dense (this can be thought of as an uncertainty principle, as in [22]).

Obviously, the complete bipartite graph cannot be directly embedded in the simplicized grid graph. However, if the grid is sufficiently large (specifically, having side length $\Omega(|\mathcal{X}|^2)$), then the complete bipartite circuit defining $\mathcal{Y}$ in terms of $\mathcal{X}$ can in fact be simulated on the simplicized grid graph. Paths between all pairs of $\mathcal{X}$ and $\mathcal{Y}$ are constructed to avoid overlap. Vertex crossings are inevitable, but they can be replaced by constant-size "swap gadgets" which simulate crossing paths via the XOR/addition swapping trick:

$$x := x + y; \qquad y := x - y; \qquad x := x - y.$$

See Figure 3 for a schematic of the implementation on the grid graph (not showing swap gadgets).

Unminoring: With the above techniques, we can prove that there is a precision matrix Θ supported on the simplicized grid graph, such that for any preconditioner, the preconditioned Lasso needs a polynomial number of samples to succeed when covariates are drawn from $N(0, \Theta^{-1})$. To extend this result, we make the following simple observation: if a covariance matrix Σ can be S-preconditioned so that Lasso succeeds at sparse recovery with covariates from $N(0, \Sigma)$, then certainly the same holds for any submatrix Σ_{VV}; the preconditioner is just S_V. In the language of precision matrices, this means that a precision matrix Θ is a hard instance (against all preconditioners) if it has a Schur complement $\Theta/\Theta_{\bar{V}\bar{V}}$ which is a hard instance.

Our goal is therefore to prove that for any high-treewidth graph G, there is a precision matrix Θ supported on G and a vertex subset V such that the Schur complement $\Theta/\Theta_{\bar{V}\bar{V}}$ approximates the hard simplicized grid instance. We appeal to the celebrated Grid Minor Theorem, which states that any graph with treewidth t contains a grid minor of size $t^{\Omega(1)} \times t^{\Omega(1)}$ [13], [15]. Finally, we prove that if G, H are graphs and H is a minor of G, then any positive-definite matrix supported on H can be approximated to arbitrary accuracy as a Schur complement of some positive-definite matrix supported on G. This last step is technically involved, but the construction is fairly simple: since H is a minor of G, each vertex of H corresponds to a connected component of G, and any edge in H corresponds to an edge between the respective components in G. Given a matrix Γ supported on H, we construct a nearly block-diagonal matrix Θ on G, where each block is a large multiple of the Laplacian of the induced subgraph of a component of G. This means each

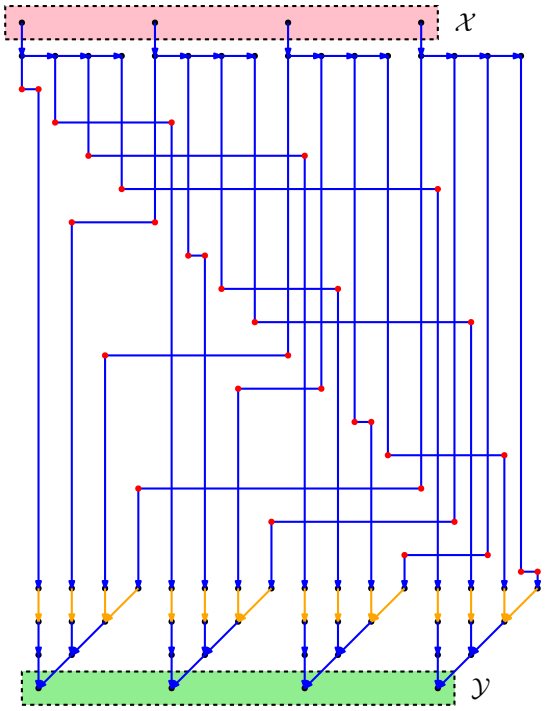

Figure 3: Schematic of equations defining the kernel of a positive-semidefinite matrix $\tilde{\Theta}$ supported on the simplicized grid graph. The kernel has dimension $|\mathcal{X}|$, and every vector in the kernel (i.e. solution to the equations) is robustly dense on either $\mathcal{X}$ or $\mathcal{Y}$. Note that there is a path from every vertex of $\mathcal{X}$ to every vertex of $\mathcal{Y}$. This allows us to enforce the constraint that for any solution, each variable of $\mathcal{Y}$ is equal to a random linear combination of the variables of $\mathcal{X}$. Every blue directed path denotes that all vertices on the path are constrained to be equal. A vertex with multiple incoming arrows indicates that the vertex is constrained to be the sum of its predecessors. The orange arrows are assigned Gaussian random weights. Every crossing between two paths is replaced by a "swap gadget" ensuring that the paths do not interfere. Vertices not on any path are constrained to be 0.

block is approximately a *Gaussian free field* [58], which induces a strong positive correlation between the variables inside the block. The entries of Γ are then assigned to appropriate edges of G and added into Θ.

Using this result, we extend our lower bounds against preconditioned Lasso to all high-treewidth graphs, completing our tight graphical characterization of the power of preconditioned Lasso.

III. Further Related Work

Generalizations of the Lasso: There is an immense literature on generalizations of the Lasso. However, to our knowledge, our work is the first to study the preconditioned Lasso as defined above, for the purpose of solving sparse linear regression. It is worth contrasting with two related branches of prior work:

1) The *generalized Lasso* [57], [60] is defined as
$$\operatorname*{argmin}_{w\in\mathbb{R}^n} \|Y - Xw\|_2^2 + \lambda \|Dw\|_1 ,$$
for a penalty matrix D. Definition I.1 is a certainly a program of this form. However, the motivation is quite different: while we consider the preconditioned Lasso as a class of approaches for linear regression with *sparsity in the original basis*, the generalized Lasso was introduced to encapsulate "problems that use the ℓ_1 norm to enforce certain structural constraints—instead of pure sparsity" [60]. That line of work has largely focused on algorithms for the generalized Lasso and applications for specific choices of penalty matrix D.
2) A different notion of preconditioned Lasso introduced in prior work [67], [37] is the notion of solving the Lasso on "preconditioned" samples (AX, AY), for some invertible matrix A, instead of the original samples (X, Y). This approach has the same motivation as ours: modifying the problem so that Lasso will succeed at signal recovery (or e.g. sign recovery [37]) for sparse linear regression. However, the two kinds of preconditioning (left vs right) are very different: theirs occurs in the space of samples, whereas ours occurs in the space of parameters. This is most clear in the noiseless setting, where Lasso reduces (if we send $\lambda \to 0$) to the basis pursuit program
$$\operatorname*{argmin}_{w\in\mathbb{R}^n : Xw=Y} \|w\|_1 .$$
As defined in [37], preconditioning has no effect on the basis pursuit program. In contrast, with our definition, preconditioned basis pursuit can often provably succeed where basis pursuit fails, as exhibited in Section I-B. We do note that [67] also suggested a more general definition of preconditioning which includes ours, though they focused on the effect of left preconditioning as discussed above.

Lower bound related work: Impossibility results for the sparse linear regression problem fall into several categories, depending on whether they address the fixed-design setting or the random design setting, and on what classes of algorithms they rule out. In the fixed design setting, there are computational lower bounds against finding a sparse solution to a system of linear equations [47], [69], [26], [33]. No comparable results are known for random designs; there is a lower bound for *robust* sparse linear regression under an assumption related to hardness of planted clique [8], but this appears to be an unrelated phenomenon, in that it holds even when Σ is the identity matrix.

There is a richer literature on lower bounds specifically against the Lasso, for both fixed and random designs. In

the random design setting, most focus on well-conditioned or identity covariances, and seek to pinpoint the constant factor in sample $m = ck \log n$ [64], [14], [52]. In contrast, we seek asymptotically stronger sample complexity lower bounds, which of course requires passing to ill-conditioned covariance matrices. There is also prior work bounding what rates the Lasso can achieve, in terms of the compatibility constant of the design matrix and the regularization parameter λ [61], [3], [4]. However, these results in general provide no lower bounds against Lasso in the noiseless setting, where the tuning parameter λ is sent to 0. Moreover, these works do not touch upon the issue of preconditioning.

In that vein, our work is most closely related to [70], which constructs a (fixed-design) lower bound against the generalization of Lasso to arbitrary coordinate-separable penalties instead of the ℓ_1 norm. Analogous to our motivation, they considered this class because there are problems for which the Lasso fails, but succeeds after an appropriate diagonal preconditioning. Note that when the penalty is a weighted linear combination of the magnitudes of the regression coefficients, this corresponds to a diagonal preconditioner. However, coordinate-separable penalties do not encompass the full power of preconditioning by an arbitrary matrix (and vice versa). Indeed, it is a limitation of the prior work that the constructed lower bound design matrices are block-diagonal with block size 2, and therefore amenable to being solved by the preconditioned Lasso. This is a limitation we address in our work.

IV. Conclusion

Our results give an answer to the question of when preconditioning the Lasso can make sparse linear regression problems tractable. On the one hand, there is an efficient preconditioning algorithm when the covariates have low-treewidth dependency structure. On the other hand, low-treewidth dependency structures are the only dependency structures which enable preconditioning: i.e., any high-treewidth dependency structure admits covariates which cannot be preconditioned.

For future work, it would be interesting to prove lower bounds for sparse linear regression against an even larger class of algorithms, and we expect some of the tools developed in this work may be useful in this direction. Conversely, it would be interesting if sparse linear regression is in fact tractable on the random designs we constructed. There is a notable lack of algorithms which succeed outside the regime of preconditioned Lasso, so it seems likely that this would require developing new algorithmic techniques.

Acknowledgements

We thank Ankur Moitra, Pablo Parrilo, Arsen Vasilyan, Philippe Rigollet, Guy Bresler, Dylan Foster, Tselil Schramm, and Matthew Brennan for valuable conversations on related topics. We also thank the anonymous reviewers for their valuable comments.

References

[1] Benny Applebaum, Boaz Barak, and David Xiao. On basing lower-bounds for learning on worst-case assumptions. In *2008 49th Annual IEEE Symposium on Foundations of Computer Science*, pages 211–220. IEEE, 2008.

[2] Richard G Baraniuk, Volkan Cevher, Marco F Duarte, and Chinmay Hegde. Model-based compressive sensing. *IEEE Transactions on information theory*, 56(4):1982–2001, 2010.

[3] Pierre C Bellec. The noise barrier and the large signal bias of the lasso and other convex estimators. *arXiv preprint arXiv:1804.01230*, 2018.

[4] Pierre C Bellec, Guillaume Lecué, Alexandre B Tsybakov, et al. Slope meets lasso: improved oracle bounds and optimality. *Annals of Statistics*, 46(6B):3603–3642, 2018.

[5] Peter J Bickel, Ya'acov Ritov, Alexandre B Tsybakov, et al. Simultaneous analysis of lasso and dantzig selector. *The Annals of statistics*, 37(4):1705–1732, 2009.

[6] Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006.

[7] Thomas Blumensath and Mike E Davies. Iterative hard thresholding for compressed sensing. *Applied and computational harmonic analysis*, 27(3):265–274, 2009.

[8] Matthew Brennan and Guy Bresler. Reducibility and statistical-computational gaps from secret leakage. In *Conference on Learning Theory*, pages 648–847. PMLR, 2020.

[9] Peter J Brockwell and Richard A Davis. *Time series: theory and methods*. Springer science & business media, 2009.

[10] Emmanuel Candes, Terence Tao, et al. The dantzig selector: Statistical estimation when p is much larger than n. *Annals of statistics*, 35(6):2313–2351, 2007.

[11] Emmanuel J Candès, Justin Romberg, and Terence Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on information theory*, 52(2):489–509, 2006.

[12] Emmanuel J Candes and Terence Tao. Decoding by linear programming. *IEEE transactions on information theory*, 51(12):4203–4215, 2005.

[13] Chandra Chekuri and Julia Chuzhoy. Polynomial bounds for the grid-minor theorem. *Journal of the ACM (JACM)*, 63(5):1–65, 2016.

[14] Xi Chen, Adityanand Guntuboyina, and Yuchen Zhang. On bayes risk lower bounds. *The Journal of Machine Learning Research*, 17(1):7687–7744, 2016.

[15] Julia Chuzhoy and Zihan Tan. Towards tight (er) bounds for the excluded grid theorem. *Journal of Combinatorial Theory, Series B*, 146:219–265, 2021.

[16] Arnak S Dalalyan, Mohamed Hebiri, Johannes Lederer, et al. On the prediction performance of the lasso. *Bernoulli*, 23(1):552–581, 2017.

[17] Abhimanyu Das and David Kempe. Algorithms for subset selection in linear regression. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 45–54, 2008.

[18] Abhimanyu Das and David Kempe. Submodular meets spectral: Greedy algorithms for subset selection, sparse approximation and dictionary selection. *arXiv preprint arXiv:1102.3975*, 2011.

[19] Arthur P Dempster. Covariance selection. *Biometrics*, pages 157–175, 1972.

[20] Sally Dong, Yin Tat Lee, and Guanghao Ye. A nearly-linear time algorithm for linear programs with small treewidth: A multiscale representation of robust central path. *arXiv preprint arXiv:2011.05365*, 2020.

[21] David L Donoho. Compressed sensing. *IEEE Transactions on information theory*, 52(4):1289–1306, 2006.

[22] David L Donoho and Philip B Stark. Uncertainty principles and signal recovery. *SIAM Journal on Applied Mathematics*, 49(3):906–931, 1989.

[23] Ethan R Elenberg, Rajiv Khanna, Alexandros G Dimakis, Sahand Negahban, et al. Restricted strong convexity implies weak submodularity. *Annals of Statistics*, 46(6B):3539–3568, 2018.

[24] Jianqing Fan, Jinchi Lv, and Lei Qi. Sparse high-dimensional models in economics. 2011.

[25] Uriel Feige, MohammadTaghi Hajiaghayi, and James R Lee. Improved approximation algorithms for minimum weight vertex separators. *SIAM Journal on Computing*, 38(2):629–657, 2008.

[26] Dean Foster, Howard Karloff, and Justin Thaler. Variable selection is hard. In *Conference on Learning Theory*, pages 696–709. PMLR, 2015.

[27] Rina Foygel and Nathan Srebro. Fast rate and optimistic rate for l1-regularized regression. Technical report, Toyota Technological Institute. arXiv: 1108.037 v1, 2011.

[28] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, 2008.

[29] David Gamarnik and Ilias Zadik. Sparse high-dimensional linear regression. algorithmic barriers and a local search algorithm. *arXiv preprint arXiv:1711.04952*, 2017.

[30] Alan George. Nested dissection of a regular finite element mesh. *SIAM Journal on Numerical Analysis*, 10(2):345–363, 1973.

[31] Leonidas Guibas, John Hershberger, Daniel Leven, Micha Sharir, and Robert E Tarjan. Linear-time algorithms for visibility and shortest path problems inside triangulated simple polygons. *Algorithmica*, 2(1-4):209–233, 1987.

[32] Alfred Haar. Zur theorie der orthogonalen funktionensysteme. *Mathematische Annalen*, 71(1):38–53, 1911.

[33] Sariel Har-Peled, Piotr Indyk, and Sepideh Mahabadi. Approximate sparse linear regression. *arXiv preprint arXiv:1609.08739*, 2016.

[34] Haitham Hassanieh, Piotr Indyk, Dina Katabi, and Eric Price. Simple and practical algorithm for sparse fourier transform. In *Proceedings of the twenty-third annual ACM-SIAM symposium on Discrete Algorithms*, pages 1183–1194. SIAM, 2012.

[35] Jan-Christian Hütter and Philippe Rigollet. Optimal rates for total variation denoising. In *Conference on Learning Theory*, pages 1115–1146. PMLR, 2016.

[36] Prateek Jain, Ambuj Tewari, and Purushottam Kar. On iterative hard thresholding methods for high-dimensional m-estimation. *arXiv preprint arXiv:1410.5137*, 2014.

[37] Jinzhu Jia, Karl Rohe, et al. Preconditioning the lasso for sign consistency. *Electronic Journal of Statistics*, 9(1):1150–1172, 2015.

[38] Jonathan Kelner, Frederic Koehler, Raghu Meka, and Ankur Moitra. Learning some popular gaussian graphical models without condition number bounds. In *Proceedings of Neural Information Processing Systems (NeurIPS)*, 2020.

[39] Vladimir Koltchinskii and Stanislav Minsker. l_1-penalization in functional linear regression with subgaussian design. *Journal de l'École polytechnique-Mathématiques*, 1:269–330, 2014.

[40] Michael Krivelevich. The phase transition in site percolation on pseudo-random graphs. *the electronic journal of combinatorics*, 22:P00, 2015.

[41] Steffen L Lauritzen. *Graphical models*, volume 17. Clarendon Press, 1996.

[42] Steffen L Lauritzen and David J Spiegelhalter. Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society: Series B (Methodological)*, 50(2):157–194, 1988.

[43] Shlomo Levy and Peter K Fullagar. Reconstruction of a sparse spike train from a portion of its spectrum and application to high-resolution deconvolution. *Geophysics*, 46(9):1235–1243, 1981.

[44] Richard J Lipton, Donald J Rose, and Robert Endre Tarjan. Generalized nested dissection. *SIAM journal on numerical analysis*, 16(2):346–358, 1979.

[45] Stéphane Mallat. *A wavelet tour of signal processing*. Elsevier, 1999.

[46] Nicolai Meinshausen, Peter Bühlmann, et al. High-dimensional graphs and variable selection with the lasso. *The annals of statistics*, 34(3):1436–1462, 2006.

[47] Balas Kausik Natarajan. Sparse approximate solutions to linear systems. *SIAM journal on computing*, 24(2):227–234, 1995.

[48] Deanna Needell and Rachel Ward. Stable image reconstruction using total variation minimization. *SIAM Journal on Imaging Sciences*, 6(2):1035–1058, 2013.

[49] Judea Pearl. *Causality*. Cambridge university press, 2009.

[50] Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. *Elements of causal inference: foundations and learning algorithms*. The MIT Press, 2017.

[51] Garvesh Raskutti, Martin J Wainwright, and Bin Yu. Restricted eigenvalue properties for correlated gaussian designs. *The Journal of Machine Learning Research*, 11:2241–2259, 2010.

[52] Galen Reeves, Jiaming Xu, and Ilias Zadik. The all-or-nothing phenomenon in sparse linear regression. In *Conference on Learning Theory*, pages 2652–2663. PMLR, 2019.

[53] Mark Rudelson and Roman Vershynin. Sparse reconstruction by convex relaxation: Fourier and gaussian measurements. In *2006 40th Annual Conference on Information Sciences and Systems*, pages 207–212. IEEE, 2006.

[54] Yousef Saad. *Iterative methods for sparse linear systems*. SIAM, 2003.

[55] Fadil Santosa and William W Symes. Linear inversion of band-limited reflection seismograms. *SIAM Journal on Scientific and Statistical Computing*, 7(4):1307–1330, 1986.

[56] James Sharpnack, Aarti Singh, and Akshay Krishnamurthy. Detecting activations over graphs using spanning tree wavelet bases. In *Artificial Intelligence and Statistics*, pages 536–544. PMLR, 2013.

[57] Yiyuan She. Sparse regression with exact clustering. *Electronic Journal of Statistics*, 4(none):1055 – 1096, 2010.

[58] Scott Sheffield. Gaussian free fields for mathematicians. *Probability theory and related fields*, 139(3-4):521–541, 2007.

[59] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.

[60] Ryan J Tibshirani and Jonathan Taylor. The solution path of the generalized lasso. *The annals of statistics*, 39(3):1335–1371, 2011.

[61] Sara Van De Geer. On tight bounds for the lasso. *Journal of Machine Learning Research*, 19:46, 2018.

[62] Sara van de Geer, Johannes Lederer, et al. The lasso, correlated design, and improved oracle inequalities. In *From Probability to Statistics and Back: High-Dimensional Models and Processes–A Festschrift in Honor of Jon A. Wellner*, pages 303–316. Institute of Mathematical Statistics, 2013.

[63] Sara A Van De Geer, Peter Bühlmann, et al. On the conditions used to prove oracle results for the lasso. *Electronic Journal of Statistics*, 3:1360–1392, 2009.

[64] Martin J Wainwright. Sharp thresholds for high-dimensional and noisy sparsity recovery using l1-constrained quadratic programming (lasso). *IEEE transactions on information theory*, 55(5):2183–2202, 2009.

[65] Martin J Wainwright. *High-dimensional statistics: A non-asymptotic viewpoint*, volume 48. Cambridge University Press, 2019.

[66] Martin J Wainwright and Michael Irwin Jordan. *Graphical models, exponential families, and variational inference*. Now Publishers Inc, 2008.

[67] Fabian L Wauthier, Nebojsa Jojic, and Michael I Jordan. A comparative framework for preconditioned lasso algorithms. *Advances in Neural Information Processing Systems*, 26:1061–1069, 2013.

[68] Tong Tong Wu, Yi Fang Chen, Trevor Hastie, Eric Sobel, and Kenneth Lange. Genome-wide association analysis by lasso penalized logistic regression. *Bioinformatics*, 25(6):714–721, 2009.

[69] 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, 2014.

[70] Yuchen Zhang, Martin J Wainwright, Michael I Jordan, et al. Optimal prediction for sparse linear models? lower bounds for coordinate-separable m-estimators. *Electronic Journal of Statistics*, 11(1):752–799, 2017.

[71] Shuheng Zhou. Restricted eigenvalue conditions on subgaussian random matrices. *arXiv preprint arXiv:0912.4045*, 2009.