Computational and statistical tradeoffs via convex relaxation

Venkat Chandrasekaran[a] and Michael I. Jordan[b,1]

[a]Departments of Computing and Mathematical Sciences and Electrical Engineering, California Institute of Technology, Pasadena, CA 91125; and [b]Departments of Statistics and Electrical Engineering and Computer Sciences, University of California, Berkeley, CA 94720

Contributed by Michael I. Jordan, February 5, 2013 (sent for review November 27, 2012)

Modern massive datasets create a fundamental problem at the intersection of the computational and statistical sciences: how to provide guarantees on the quality of statistical inference given bounds on computational resources, such as time or space. Our approach to this problem is to define a notion of "algorithmic weakening," in which a hierarchy of algorithms is ordered by both computational efficiency and statistical efficiency, allowing the growing strength of the data at scale to be traded off against the need for sophisticated processing. We illustrate this approach in the setting of denoising problems, using convex relaxation as the core inferential tool. Hierarchies of convex relaxations have been widely used in theoretical computer science to yield tractable approximation algorithms to many computationally intractable tasks. In the current paper, we show how to endow such hierarchies with a statistical characterization and thereby obtain concrete tradeoffs relating algorithmic runtime to amount of data.

convex geometry | convex optimization | high-dimensional statistics

The rapid growth in the size and scope of datasets in science and technology has created a need for novel foundational perspectives on data analysis that blend computer science and statistics. That classical perspectives from these fields are not adequate to address emerging problems in "Big Data" is apparent from their sharply divergent nature at an elementary level: In computer science, the growth of the number of data points is a source of "complexity" that must be tamed via algorithms or hardware, whereas in statistics, the growth of the number of data points is a source of "simplicity" in that inferences are generally stronger and asymptotic results can be invoked. In classical statistics, where one considers the increase in inferential accuracy as the number of data points grows, there is little or no consideration of computational complexity. Indeed, if one imposes the additional constraint, as is prevalent in real-world applications, that a certain level of inferential accuracy must be achieved within a limited time budget, classical theory provides no guidance as to how to design an inferential strategy. [Note that classical statistics contains a branch known as *sequential analysis* that does discuss methods that stop collecting data points after a target error level has been reached (e.g., ref. 1), but this is different from the computational complexity guarantees (the number of steps that a computational procedure requires) that are our focus.] In classical computer science, practical solutions to large-scale problems are often framed in terms of approximations to idealized problems; however, even when such approximations are sought, they are rarely expressed in terms of the coin of the realm of the theory of inference: the statistical risk function. Thus, there is little or no consideration of the idea that computation can be simplified in large datasets because of the enhanced inferential power in the data. In general, in computer science, datasets are not viewed formally as a resource on a par with time and space (such that the more of the resource, the better).

On intuitive grounds, it is not implausible that strategies can be designed to yield monotonically improving risk as data accumulate, even in the face of a time budget. In particular, if an algorithm simply ignores all future data once a time budget is exhausted, statistical risk will not increase (under various assumptions that may not be desirable in practical applications). Alternatively, one might allow linear growth in the time budget (e.g., in a real-time setting) and attempt to achieve such growth via a subsampling strategy in which some fraction of the data are dropped. Executing such a strategy may be difficult, however, in that the appropriate fraction depends on the risk function, and thus on a mathematical analysis that may be difficult to carry out. Moreover, subsampling is a limited strategy for controlling computational complexity. More generally, one would like to consider some notion of "algorithm weakening," where as data accumulate, one can back off to simpler algorithmic strategies that nonetheless achieve a desired risk. The challenge is to do this in a theoretically sound manner.

We base our approach to this problem on the notion of a "time-data complexity class." In particular, we define a class $\mathbb{TD}(t(p), n(p), \varepsilon(p))$ of parameter estimation problems in which a p-dimensional parameter underlying an unknown population can be estimated with a risk of $\varepsilon(p)$, given $n(p)$ independent and identically distributed (i.i.d.) samples using an inference procedure with runtime $t(p)$. Our definition parallels the definition of the time-space (TISP) complexity class in computational complexity theory for describing algorithmic tradeoffs between time and space resources (2). In this formalization, classical results in estimation theory can be viewed as emphasizing the tradeoffs between the second and third parameters (amount of data and risk). Our focus in this paper is to fix $\varepsilon(p)$ to some desired level of accuracy and to investigate the tradeoffs between the first two parameters, namely, runtime and dataset size.

Although classical statistics gave little consideration to computational complexity, computational issues have come increasingly to the fore in modern "high-dimensional statistics" (3), where the number of parameters p is relatively large and the number of data points n is relatively small. In this setting, methods based on convex optimization have been emphasized (particularly methods based on ℓ_1 penalties). This is due, in part, to the favorable analytical properties of convex functions and convex sets, but also to the fact that such methods tend to have favorable computational scaling. However, the treatment of computation has remained in-

Significance

The growth in the size and scope of datasets in science and technology has created a need for foundational perspectives on data analysis that blend computer science and statistics. Specifically, the core challenge with massive datasets is that of guaranteeing improved accuracy of an analysis procedure as data accrue, even in the face of a time budget. We address this problem via a notion of "algorithmic weakening," whereby as data scale, the procedure backs off to cheaper algorithms, leveraging the growing inferential strength of the data to ensure that a desired level of accuracy is achieved within the computational budget.

Author contributions: V.C. and M.I.J. designed research, performed research, and wrote the paper.

The authors declare no conflict of interest.

Freely available online through the PNAS open access option.

[1]To whom correspondence should be addressed. E-mail: jordan@eecs.berkeley.edu.

This article contains supporting information online at www.pnas.org/lookup/suppl/doi:10.1073/pnas.1302293110/-/DCSupplemental.

Downloaded from https://www.pnas.org by 103.238.106.114 on December 13, 2022 from IP address 103.238.106.114.

Downloaded from https://www.pnas.org by 103.238.106.114 on December 13, 2022 from IP address 103.238.106.114.

formal, with little attempt to characterize tradeoffs between computation time and estimation quality. In our work, we aim explicitly at such tradeoffs in the setting in which both n and p are large.

To develop a notion of algorithm weakening that combines computational and statistical considerations, we consider estimation procedures for which we can characterize the computational benefits as well as the loss in estimation performance due to the use of weaker algorithms. Reflecting the fact that the space of all algorithms is poorly understood, we retain the focus on convex optimization from high-dimensional statistics, but we consider parameterized hierarchies of optimization procedures in which a form of algorithm weakening is obtained by using successively weaker outer approximations to convex sets. Such *convex relaxations* have been widely used to give efficient approximation algorithms for intractable problems in computer science (4). As we will discuss, a precise characterization of both the estimation performance and the computational complexity of using a particular relaxation of a convex set can be obtained by appealing to convex geometry and to results on the complexity of solving convex programs. Specifically, the tighter relaxations in these families offer better approximation quality (and better estimation performance in our context) but are computationally more complex. On the other hand, the weaker relaxations are computationally more tractable and can provide the same estimation performance as the tighter ones but with access to more data. In this manner, convex relaxations provide a principled mechanism to weaken inference algorithms so as to reduce the runtime in processing larger datasets.

To demonstrate explicit tradeoffs in high-dimensional, large-scale inference, we focus, for simplicity and concreteness, on estimation in sequence models (5):

$$\mathbf{y} = \mathbf{x}^* + \sigma\mathbf{z}, \qquad [1]$$

where $\sigma > 0$, the noise vector $\mathbf{z} \in \mathbb{R}^p$ is standard normal, and the unknown parameter $\mathbf{x}^*$ belongs to a known subset $\mathcal{S} \subset \mathbb{R}^p$. The objective is to estimate $\mathbf{x}^*$ based on n independent observations $\{\mathbf{y}_i\}_{i=1}^n$ of $\mathbf{y}$. This denoising setup has a long history and has been at the center of some remarkable results in the high-dimensional setting over the past two decades, beginning with the papers of Donoho and Johnstone (6, 7). The estimators discussed next proceed by first computing the sample mean $\bar{\mathbf{y}} = \sum_{i=1}^n \mathbf{y}_i$ and then using $\bar{\mathbf{y}}$ as input to a suitable convex program. Of course, this is equivalent to a denoising problem in which the noise variance is σ^2/n, and we are given just one sample. The reason why we consider the elaborate two-step procedure is to account more accurately both for data aggregation and for subsequent processing in our runtime calculations. Indeed, in a real-world setting, one is typically faced with a massive dataset in unaggregated form, and when both p and n may be large, summarizing the data before any further processing can itself be an expensive computation. As will be seen in concrete calculations of time-data tradeoffs, the number of operations corresponding to data aggregation is sometimes comparable to or even larger than the number of operations required for subsequent processing in a massive data setting.

To estimate $\mathbf{x}^*$, we consider the following natural shrinkage estimator given by a projection of the sample mean $\bar{\mathbf{y}}$ onto a convex set $\mathcal{C}$ that is an outer approximation to $\mathcal{S}$ (i.e., $\mathcal{S} \subset \mathcal{C}$):

$$\hat{\mathbf{x}}_n(\mathcal{C}) = \arg\min_{\mathbf{x}\in\mathbb{R}^p} \ \frac{1}{2}\left\|\bar{\mathbf{y}} - \mathbf{x}\right\|_{\ell_2}^2 \ \text{ s.t. } \mathbf{x} \in \mathcal{C}. \qquad [2]$$

We study the estimation performance of a family of shrinkage estimators $\{\hat{\mathbf{x}}_n(\mathcal{C}_i)\}$ that use as the convex constraint one of a sequence of convex outer approximations $\{\mathcal{C}_i\}$ with $\mathcal{C}_1 \supset \mathcal{C}_2 \supset \ldots \supset \mathcal{S}$. Given the same number of samples, using a weaker relaxation, such as $\mathcal{C}_1$, leads to an estimator with a larger risk than would result from using a tighter relaxation, such as $\mathcal{C}_2$. On the other hand, given access to more data samples, the weaker approximations provide the same estimation guarantees as the tighter ones. In settings in which computing a weaker approximation is more tractable than computing a tighter one, a natural computation/sample tradeoff arises. We characterize this tradeoff in a number of stylized examples, motivated by problems such as collaborative filtering, learning an ordering of a collection of random variables, and inference in networks.

More broadly, this paper highlights the role of computation in estimation by jointly studying both the computational and statistical aspects of high-dimensional inference. Such an understanding is particularly of interest in modern inferential tasks in data-rich settings. Furthermore, an observation from our examples on time-data tradeoffs is that, in many contexts, one does not need too many extra data samples to go from a computationally inefficient estimator based on a tight relaxation to an extremely efficient estimator based on a weaker relaxation. Consequently, in application domains in which obtaining more data is not too expensive, it may be preferable to acquire more data, with the upshot being that the computational infrastructure can be relatively less sophisticated.

We should note that we investigate only one algorithm-weakening mechanism, namely, convex relaxation, and one class of statistical estimation problems, namely, denoising in a high-dimensional sequence model. There is reason to believe, however, that the principles described in this paper are relevant more generally. Convex optimization-based procedures are used in a variety of large-scale data analysis tasks (3, 8), and it is likely to be interesting to explore hierarchies of convex relaxations in such tasks. In addition, there are a number of potentially interesting mechanisms beyond convex relaxation for weakening inference procedures, such as dimensionality reduction or other forms of data quantization and approaches based on clustering or coresets. We discuss these and other research directions in *Conclusions*.

Related Work

A number of papers have considered computational and sample complexity tradeoffs in the setting of learning binary classifiers. Specifically, several authors have described settings under which speedups in running time of a classifier learning algorithm are possible, given a substantial increase in dataset size (9–12). In contrast, in the denoising setup considered in this paper, several of our examples of time-data tradeoffs demonstrate significant computational speedups with just a constant factor increase in dataset size. Another attempt in the binary classifier learning setting, building on earlier work on classifier learning in data-rich problems (13), has shown that modest improvements in runtime (of constant factors) may be possible with access to more data by using the stochastic gradient descent method (14). Time-data tradeoffs have also been characterized in Boolean network training from time series data (15), but the computational speedups offered there are from exponential-time algorithms to slightly faster but still exponential-time algorithms. Two recent papers (16, 17) have considered time-data tradeoffs in sparse principal component analysis (PCA) (18) and in biclustering, in which one wishes to estimate the support of a sparse eigenvector that consists of most of the energy of a matrix. We also study time-data tradeoffs for this problem, but from a denoising perspective rather than from one of estimating the support of the leading sparse eigenvector. In our discussion of *Example 3* (*Time-Data Tradeoffs*), we discuss the differences between our problem setup and these latter two papers (16, 17). Finally, a recent paper (19) studies time-data tradeoffs in model selection problems by investigating procedures that operate within a computational budget. As a general contrast to all these previous results, a major contribution of the present paper is the demonstration of the efficacy of convex relaxation as a powerful algorithm-weakening mechanism for processing massive datasets in a broad range of settings.

Paper Outline

The main sections of this paper proceed in the following sequence. The next section describes a framework for formally stating results on time-data tradeoffs. We then provide some background on convex optimization and relaxations of convex sets. Following this, we investigate in detail the denoising problem [**1**] and char-

Downloaded from https://www.pnas.org by 103.238.106.114 on December 13, 2022 from IP address 103.238.106.114.

acterize the risk obtained when one employs a convex programming estimator of the type [**2**]. Subsequently, we give several examples of time-data tradeoffs in concrete denoising problems. Finally, we conclude with a discussion of directions for further research.

Formally Stating Time-Data Tradeoffs

In this section, we describe a framework to state results on computational and statistical tradeoffs in estimation. Our discussion is relevant to general parameter estimation problems and inference procedures; one may keep in mind the denoising problem [**1**] for concreteness. Consider a sequence of estimation problems indexed by the dimension p of the parameter to be estimated. Fix a risk function $\varepsilon(p)$ that specifies the desired error of an estimator. For example, in the denoising problem [**1**], the error of an estimator of the form [**2**] may be specified as the worst case mean squared error taken over all elements of the set $\mathcal{S}$ (i.e., $\sup_{\mathbf{x}^* \in \mathcal{S}} \mathbb{E}[\|\mathbf{x}^* - \hat{\mathbf{x}}_n(\mathcal{C})\|_{\ell_2}^2]$).

One can informally view an estimation algorithm that achieves a risk of $\varepsilon(p)$ by processing $n(p)$ samples with runtime $t(p)$ as a point on a 2D plot as shown in Fig. 1, with one axis representing the runtime and the other representing the sample complexity. To be precise, the axes in the plot index *functions* (of p) that represent runtime and number of samples, but we do not emphasize such formalities and rather use these plots to provide a useful qualitative comparison of inference algorithms. In Fig. 1, procedure A requires fewer samples than procedure C to achieve the same error, but this reduction in sample complexity comes at the expense of a larger runtime. Procedure B has both a larger sample complexity and a larger runtime than procedure C; thus, it is strictly dominated by procedure C.

Given an error function $\varepsilon(p)$, there is a lower bound on the number of samples $n(p)$ required to achieve this error using any computational procedure [i.e., no constraints on $t(p)$]; such information-theoretic or minimax risk lower bounds correspond to "vertical lines" in the plot in Fig. 1. Characterizing these fundamental limits on sample complexity has been a traditional focus in the estimation theory literature, with a fairly complete set of results available in many settings. One can imagine asking for similar lower bounds on the computational side, corresponding to "horizontal lines" in the plot in Fig. 1: Given a desired risk $\varepsilon(p)$ and access to an unbounded number of samples, what is a nontrivial lower bound on the runtime $t(p)$ of any inference algorithm that achieves a risk of $\varepsilon(p)$? Such complexity-theoretic lower bounds are significantly harder to obtain, and they remain a central open problem in computational complexity theory.

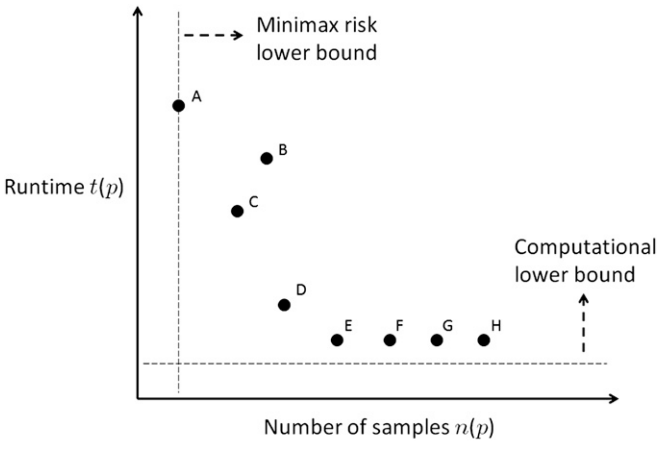

Fig. 1. Tradeoff between the runtime and sample complexity in a stylized parameter estimation problem. Here, the risk is assumed to be fixed to some desired level, and the points in the plot refer to different algorithms that require a certain runtime and a certain number of samples to achieve the desired risk. The vertical and horizontal lines refer to lower bounds in sample complexity and in runtime, respectively.

This research landscape informs the qualitative nature of the statements on time-data tradeoffs we make in this paper. First, we will not attempt to prove combined lower bounds, as is traditionally done in the characterization of tradeoffs between physical quantities, involving $n(p)$ and $t(p)$ jointly; this is because obtaining a lower bound just on $t(p)$ remains a substantial challenge. Hence, our time-data tradeoff results on the use of more efficient algorithms for larger datasets refer to a reduction in the *upper bounds* on runtimes of estimation procedures with increases in dataset size. Second, in any setting in which there is a computational cost associated with touching each data sample and in which the samples are exchangeable, there is a sample threshold beyond which it is computationally more efficient to throw away excess data samples than to process them in any form. This observation suggests that there is a "floor," as in Fig. 1 with procedures E, F, G, and H, beyond which additional data do not lead to a reduction in runtime. Precisely characterizing this sample threshold is generally very hard because it depends on difficult-to-obtain computational lower bounds for estimation tasks as well as on the particular space of estimation algorithms that one may use. We will comment further on this point when we consider concrete examples of time-data tradeoffs.

To state our results concerning time-data tradeoffs formally, we define a resource class constrained by runtime and sample complexity as follows.

Definition 1: Consider a sequence of parameter estimation problems indexed by the dimension p of the space of parameters that index an underlying population. This sequence of estimation problems belongs to a time-data class $\mathbb{TD}(t(p), n(p), \varepsilon(p))$ if there exists an inference procedure for the sequence of problems with runtime upper-bounded by $t(p)$, with the number of i.i.d. samples processed bounded by $n(p)$, and which achieves a risk bounded by $\varepsilon(p)$.

We note that our definition of a time-data resource class parallels the time-space resource classes considered in complexity theory (2). In that literature, TISP($t(p)$, $s(p)$) denotes a class of problems of input size p that can be solved by some algorithm using $t(p)$ operations and $s(p)$ units of space.

With this formalism, classical minimax bounds can be stated as follows. Given some function $\bar{n}(p)$ for the number of samples, suppose a parameter estimation problem has a minimax risk of $\varepsilon_{\text{minimax}}(p)$ [which depends on the function $\bar{n}(p)$]. If an estimator achieving a risk of $\varepsilon_{\text{minimax}}(p)$ is *computable* with runtime $\bar{t}(p)$, this estimation problem then lies in $\mathbb{TD}(\bar{t}(p), \bar{n}(p), \epsilon_{\text{minimax}}(p))$. Thus, the emphasis is fundamentally on the relationship between $\bar{n}(p)$ and $\varepsilon_{\text{minimax}}(p)$, without much focus on the computational procedure that achieves the minimax risk bound. Our interest in this paper is to fix the risk $\varepsilon(p) = \varepsilon_{\text{desired}}(p)$ to be equal to some desired level of accuracy and to investigate the tradeoffs between $t(p)$ and $n(p)$ so that a parameter estimation problem lies in $\mathbb{TD}(t(p), n(p), \varepsilon_{\text{desired}}(p))$.

Convex Relaxation

In this section, we describe the particular algorithmic toolbox on which we focus, namely, convex programs. Convex optimization methods offer a powerful framework for statistical inference due to the broad class of estimators that can be effectively modeled as convex programs. Furthermore, the theory of convex analysis is useful both for characterizing the statistical properties of convex programming-based estimators and for developing methods to compute such estimators efficiently. Most importantly from our viewpoint, convex optimization methods provide a principled and general framework for algorithm weakening based on relaxations of convex sets. We briefly discuss the key ideas from this literature that are relevant to this paper in this section. A central notion to the geometric viewpoint adopted in this section is that of a *convex cone*, which is a convex set that is closed under nonnegative linear combinations.

Representation of Convex Sets. Convex programs refer to a class of optimization problems in which we seek to minimize a convex

Downloaded from https://www.pnas.org by 103.238.106.114 on December 13, 2022 from IP address 103.238.106.114.

function over a convex constraint set (8). For example, linear programming (LP) and semidefinite programming (SDP) are two prominent subclasses in which linear functions are minimized over constraint sets given by affine spaces intersecting the nonnegative orthant (in LP) and the positive semidefinite cone (in SDP). Roughly speaking, convex programs are tractable to solve computationally if the convex objective function can be computed efficiently and if membership of an arbitrary point in the convex constraint sets can be certified efficiently; we will informally refer to this latter operation as computing the convex constraint set. (More precisely, one requires an efficient *separation oracle* that responds YES if the point is in the convex set and otherwise provides a hyperplane that separates the point from the convex set). It is then clear that the main computational bottleneck associated with solving convex programs of the form [**2**] is the efficiency of computing the constraint sets.

A central insight from the literature on convex optimization is that the complexity of computing a convex set is closely linked to how efficiently the set can be *represented*. Specifically, if a convex set can be expressed as the intersection of a small number of "basic" or "elementary" convex sets, each of which is tractable to compute, the original convex set is then also tractable to compute, and one can, in turn, optimize over this set efficiently. Examples of basic convex sets include affine spaces or cones, such as the nonnegative orthant and the cone of positive semidefinite matrices. Indeed, a canonical method with which to represent a convex set is to express the set as the intersection of a cone and an affine space. In what follows, we will consider such *conic representations* of convex sets in $\mathbb{R}^p$.

Definition 2: Let $\mathcal{C} \in \mathscr{R}^p$ be a convex set, and let $\mathcal{K} \in \mathscr{R}^p$ be a convex cone. $\mathcal{C}$ is then said to be $\mathcal{K}$-representable if $\mathcal{C}$ can be expressed as follows for $A \in \mathscr{R}^{m \times p}$, $b \in \mathscr{R}^m$:

$$\mathcal{C} = \{\mathbf{x} | \mathbf{x} \in \mathcal{K}, A\mathbf{x} = b\}. \qquad [\mathbf{3}]$$

Such a representation of $\mathcal{C}$ is called a $\mathcal{K}$ -representation.

Informally, if $\mathcal{K}$ is the nonnegative orthant (or the semidefinite cone), we will refer to the resulting representations as LP representations (or SDP representations), following commonly used terminology in the literature. A virtue of conic representations of convex sets based on the orthant or the semidefinite cone is that these representations lead to a numerical recipe for solving convex optimization problems of the form [**2**] via a natural associated barrier penalty (20). The computational complexity of these procedures is polynomial in the dimension of the cone, and we discuss runtimes for specific instances in our discussion of concrete examples of time-data tradeoffs.

Example 1: The p-dimensional *simplex* is an example of an LP representable set:

$$\Delta_p = \{\mathbf{x} | \mathbf{1}'\mathbf{x} = 1, \mathbf{x} \ge 0\}, \qquad [\mathbf{4}]$$

where $\mathbf{1} \in \mathscr{R}^p$ is the all-ones vector.

The p-simplex is the set of probability vectors in $\mathbb{R}^p$. The next example is one of an SDP-representable set that is commonly encountered both in optimization and in statistics.

Example 2: The *elliptope*, or the set of correlation matrices, in the space of $m \times m$ symmetrical matrices is defined as follows:

$$\mathcal{E}_{m \times m} = \{\mathbf{X} | \mathbf{X} \succeq 0, \mathbf{X}_{ii} = 1 \ \forall i\}. \qquad [\mathbf{5}]$$

Conic representations are somewhat limited in their modeling capacity, and an important generalization is obtained by considering *lifted* representations. In particular, the notion of *lift-and-project* plays a critical role in many examples of efficient representations of convex sets. The lift-and-project concept is simple: We wish to express a convex set $\mathcal{C} \in \mathbb{R}^p$ as the projection of a convex set $\mathcal{C}' \in \mathbb{R}^{p'}$ in some higher dimensional space (i.e., $p' > p$). The complexity of solving the associated convex programs is now a function of the lifting dimension p'. Thus, lift-and-project techniques are useful if p' is not too much larger than p and if $\mathcal{C}'$ has an efficient representation in the higher dimensional space $\mathbb{R}^{p'}$. Lift-and-project provides a very powerful representation tool, as seen in the following example.

Example 3: The *cross-polytope* is the unit ball of the ℓ_1-norm:

$$B^p_{\ell_1} = \left\{\mathbf{x} \in \mathbb{R}^p \,\middle|\, \sum_i |\mathbf{x}_i| \le 1\right\}.$$

The ℓ_1-norm has been the focus of much attention recently in statistical model selection and feature selection due to its sparsity-inducing properties (21, 22). Although the cross-polytope has *2p* vertices, a direct specification in terms of linear constraints involves 2^p inequalities:

$$B^p_{\ell_1} = \left\{\mathbf{x} \in \mathbb{R}^p \,\middle|\, \sum_i \mathbf{z}_i \mathbf{x}_i \le 1, \forall \mathbf{z} \in \{-1, +1\}^p\right\}.$$

However, we can obtain a tractable representation by lifting to $\mathscr{R}^{2p}$ and then projecting onto the first p coordinates:

$$B^p_{\ell_1} = \left\{\mathbf{x} \in \mathbb{R}^p \,|\, \exists \mathbf{z} \in \mathbb{R}^p \ \text{ s.t. } -\mathbf{z}_i \le \mathbf{x}_i \le \mathbf{z}_i \ \ \forall i, \sum_i \mathbf{z}_i \le 1\right\}.$$

Note that in $\mathscr{R}^{2p}$ with the additional variables $\mathbf{z}$, we have only $2p+1$ inequalities.

Another example of a polytope that requires many inequalities in a direct description is the *permutahedron* (23), the convex hull of all the permutations of the vector $[1, \ldots, p]' \in \mathbb{R}^p$. In fact, the permutahedron requires exponentially many linear inequalities in a direct description, whereas a lifted representation involves $\mathcal{O}(p \log(p))$ additional variables and about $\mathcal{O}(p \log(p))$ inequalities in the higher dimensional space (24). We refer the reader to the literature on conic representations for other examples (ref. 25 and references therein), including lifted semidefinite representations.

Hierarchies of Convex Relaxations. In many cases of interest, convex sets may not have tractable representations. Lifted representations in such cases have lifting dimensions that are superpolynomially large in the dimension of the original convex set; thus, the associated numerical techniques lead to intractable computational procedures that have superpolynomial runtime with respect to the dimension of the original set. A prominent example of a convex set that is difficult to compute is the *cut polytope*:

$$\mathrm{CUT}_{m \times m} = \mathrm{conv}\{\mathbf{m}\mathbf{m}' | \mathbf{m} \in \{-1, +1\}^m\}. \qquad [\mathbf{6}]$$

Rank-one signed matrices and their convex combinations are of interest in collaborative filtering and clustering problems (see *Time-Data Tradeoffs*). There is no known tractable representation of the cut polytope; lifted linear or semidefinite representations have lifting dimensions that are superpolynomial in size. Such computational issues have led to a large literature on approximating intractable convex sets by tractable ones. For the purposes of this paper, and following the dominant trend in the literature, we focus on outer approximations. For example, the elliptope [**5**] is an outer relaxation of the cut polytope, and it has been used in approximation algorithms for intractable combinatorial optimization problems, such as finding the maximum-weight cut in a graph (26). More generally, one can imagine a hierarchy of increasingly tighter approximations $\{\mathcal{C}_i\}$ of a convex set $\mathcal{C}$ as follows:

$$\mathcal{C} \subseteq \cdots \subseteq \mathcal{C}_3 \subseteq \mathcal{C}_2 \subseteq \mathcal{C}_1.$$

There exist several mechanisms for deriving such hierarchies, and we describe three frameworks here.

In the first framework, which was developed by Sherali and Adams (27), the set $\mathcal{C}$ is assumed to be polyhedral and each element of the family $\{\mathcal{C}_i\}$ is also polyhedral. Specifically, each $\mathcal{C}_i$ is expressed via a lifted LP representation. Tighter approximations are obtained by resorting to larger sized lifts such that the lifting dimension increases with the level i in the hierarchy. The second framework is similar in spirit to the first one, but the set $\mathcal{C}$ is now a convex basic, closed semialgebraic set and the approximations $\{\mathcal{C}_i\}$ are given by lifted SDP representations. [A basic, closed semialgebraic set is the collection of solutions of a

Downloaded from https://www.pnas.org by 103.238.106.114 on December 13, 2022 from IP address 103.238.106.114.

system of polynomial equations and polynomial inequalities (28).] Again, the lifting dimension increases with the level i in the hierarchy. This method was initially pioneered by Parrilo (29, 30) and by Lasserre (31), and it was studied in greater detail subsequently by Gouveia et al. (32). Both of these first and second frameworks are similar in spirit in that tighter approximations are obtained via lifted representations with successively larger lifting dimensions. The third framework that we mention here is qualitatively different from the first two. Suppose $\mathcal{C}$ is a convex set that has a $\mathcal{K}$-representation; by successively weakening the cone $\mathcal{K}$ itself, one obtains increasingly weaker approximations to $\mathcal{C}$. Specifically, we consider the setting in which the cone $\mathcal{K}$ is a hyperbolicity cone (33). Such cones have rich geometric and algebraic structure, and their boundary is given in terms of the vanishing of hyperbolic polynomials. They include the orthant and the semidefinite cone as special cases. We do not go into further technical details and formal definitions of these cones here; instead, we refer the interested reader to the work of Renegar (33). The main idea is that one can obtain a family of relaxations $\{\mathcal{K}_i\}$ to a hyperbolicity cone $\mathcal{K} \subseteq \mathbb{R}^p$, where each $\mathcal{K}_i$ is a convex cone (in fact, hyperbolic) and is a subset of $\mathbb{R}^p$:

$$\mathcal{K} \subseteq \cdots \subseteq \mathcal{K}_3 \subseteq \mathcal{K}_2 \subseteq \mathcal{K}_1. \qquad \textbf{[7]}$$

These outer conic approximations are obtained by taking certain derivatives of the hyperbolic polynomial used to define the original cone $\mathcal{K}$ (more details are provided in ref. 33). One then constructs a hierarchy of approximations $\{\mathcal{C}_i\}$ to $\mathcal{C}$ by replacing the cone $\mathcal{K}$ in the representation of $\mathcal{C}$ by the family of conic approximations $\{\mathcal{K}_i\}$. From **3** and **7**, it is clear that the approximations $\{\mathcal{C}_i\}$ so defined satisfy $\mathcal{C} \subseteq \ldots \subseteq \mathcal{C}_3 \subseteq \mathcal{C}_2 \subseteq \mathcal{C}_1$.

The important point in these three frameworks is that the family of approximations $\{\mathcal{C}_i\}$ obtained in each case is ordered both by approximation quality and by computational complexity; that is, the weaker approximations in the hierarchy are also the ones that are more tractable to compute. This observation leads to an algorithm-weakening mechanism that is useful for processing larger datasets more coarsely. As demonstrated concretely in the next section, the estimator [**2**] based on a weaker approximation to $\mathcal{C}$ can provide the same statistical performance as one based on a stronger approximation to $\mathcal{C}$, provided that the former estimator is evaluated with more data. The upshot is that the first estimator is more tractable to compute than the second. Thus, we obtain a technique for reducing the runtime required to process a larger dataset.

Estimation via Convex Optimization

In this section, we investigate the statistical properties of the estimator [**2**] for the denoising problem [**1**]. The signal set $\mathcal{S}$ in **1** differs based on the application of interest. For example, $\mathcal{S}$ may be the set of sparse vectors in a fixed basis, which could correspond to the problem of denoising sparse vectors in wavelet bases (6). The signal set $\mathcal{S}$ may be the set of low-rank matrices, which leads to problems of collaborative filtering (34). Finally, $\mathcal{S}$ may be a set of permutation matrices corresponding to rankings over a collection of items. Our analysis in this section is general, and it is applicable to these and other settings (concrete examples are provided in *Time-Data Tradeoffs*). In some denoising problems, one is interested in noise models other than Gaussian. We comment on the performance of the estimator [**2**] in settings with non-Gaussian noise, although we primarily focus on the Gaussian case for simplicity.

Convex Programming Estimators. To analyze the performance of the estimator [**2**], we introduce a few concepts from convex analysis (35). Given a closed convex set $\mathcal{C} \in \mathbb{R}^p$ and a point $\mathbf{a} \in \mathcal{C}$, we define the *tangent cone* at $\mathbf{a}$ with respect to $\mathcal{C}$ as

$$T_{\mathcal{C}}(\mathbf{a}) = \text{cone}\{\mathbf{b} - \mathbf{a} | \mathbf{b} \in \mathcal{C}\}. \qquad \textbf{[8]}$$

Here, cone$(\cdot)$ refers to the conic hull of a set obtained by taking nonnegative linear combinations of elements of the set. The cone $T_{\mathcal{C}}(\mathbf{a})$ is the set of directions to points in $\mathcal{C}$ from the point $\mathbf{a}$. The *polar* $\mathcal{K}^* \subseteq \mathbb{R}^p$ of a cone $\mathcal{K} \subseteq \mathbb{R}^p$ is the cone

$$\mathcal{K}^* = \{\mathbf{h} \in \mathbb{R}^p | \langle \mathbf{h}, \mathbf{d} \rangle \leq 0 \, \forall \, \mathbf{d} \in \mathcal{K}\}.$$

The *normal cone* $N_{\mathcal{C}}(\mathbf{a})$ at $\mathbf{a}$ with respect to the convex set $\mathcal{C}$ is the polar cone of the tangent cone $T_{\mathcal{C}}(\mathbf{a})$:

$$N_{\mathcal{C}}(\mathbf{a}) = T_{\mathcal{C}}(\mathbf{a})^*. \qquad \textbf{[9]}$$

Thus, the normal cone consists of vectors that form an obtuse angle with every vector in the tangent cone $T_{\mathcal{C}}(\mathbf{x})$. Both the tangent and normal cones are convex cones.

A key quantity that will appear in our error bounds is the following notion of the complexity or "size" of a tangent cone.

Definition 3: The Gaussian squared-complexity of a set $\mathcal{D} \in \mathbb{R}^p$ is defined as:

$$g(\mathcal{D}) = \mathbb{E}\left[\sup_{\mathbf{a} \in \mathcal{D}} \langle \mathbf{a}, \mathbf{g} \rangle^2\right],$$

where the expectation is with respect to $\mathbf{g} \sim \mathcal{N}(0, I_{p \times p})$.

This quantity is closely related to the Gaussian complexity of a set (36, 37), which consists of no squaring of the term inside the expectation. The Gaussian squared-complexity shares many properties in common with the Gaussian complexity, and we describe those that are relevant to this paper in the next subsection. Specifically, we discuss methods to estimate this quantity for sets $\mathcal{D}$ that have some structure.

With these definitions and letting $B^p_{\ell_2}$ denote the ℓ_2 ball in $\mathbb{R}^p$, we have the following result on the error between $\hat{\mathbf{x}}_n(C)$ and $\mathbf{x}^*$.

Proposition 4.

For $\mathbf{x}^* \in \mathcal{S} \subset \mathrm{R}^p$ *and with* $\mathcal{C} \subseteq \mathbb{R}^p$ *convex such that* $\mathcal{S} \subseteq \mathcal{C}$*, we have the error bound*

$$\mathbb{E}\left[\left\|\mathbf{x}^* - \hat{\mathbf{x}}_n(\mathcal{C})\right\|^2_{\ell_2}\right] \leq \frac{\sigma^2}{n} g\left(T_{\mathcal{C}}(\mathbf{x}^*) \cap B^p_{\ell_2}\right).$$

Proof: We have that $\bar{\mathbf{y}} = \mathbf{x}^* + \frac{\sigma}{\sqrt{n}}\mathbf{z}$. We begin by establishing a bound that is derived by conditioning on $\mathbf{z} = \tilde{\mathbf{z}}$. Subsequently, taking expectations concludes the proof. We have from the optimality conditions (35) of the convex program [**2**] that

$$\mathbf{x}^* + \frac{\sigma}{n}\tilde{\mathbf{z}} - \hat{\mathbf{x}}_n(\mathcal{C})|_{\mathbf{z}=\tilde{\mathbf{z}}} \in N_{\mathcal{C}}\left(\hat{\mathbf{x}}_n(\mathcal{C})|_{\mathbf{z}=\tilde{\mathbf{z}}}\right).$$

Here, $\hat{\mathbf{x}}_n(\mathcal{C})|_{\mathbf{z}=\tilde{\mathbf{z}}}$ represents the optimal value of [**2**] conditioned on $\mathbf{z} = \tilde{\mathbf{z}}$. Because $\mathbf{x}^* \in \mathcal{S} \subseteq \mathcal{C}$, we have that $\mathbf{x}^* - \hat{\mathbf{x}}_n(\mathcal{C})|_{\mathbf{z}=\tilde{\mathbf{z}}} \in T_{\mathcal{C}}(\hat{\mathbf{x}}_n(\mathcal{C})|_{\mathbf{z}=\tilde{\mathbf{z}}})$. Because the normal and tangent cones are polar to each other, we have that

$$\left\langle \mathbf{x}^* + \frac{\sigma}{\sqrt{n}}\tilde{\mathbf{z}} - \hat{\mathbf{x}}_n(\mathcal{C})|_{\mathbf{z}=\tilde{\mathbf{z}}}, \mathbf{x}^* - \hat{\mathbf{x}}_n(\mathcal{C})|_{\mathbf{z}=\tilde{\mathbf{z}}}\right\rangle \leq 0.$$

It then follows that

$$\begin{aligned}
\left\|\mathbf{x}^* - \hat{\mathbf{x}}_n(\mathcal{C})|_{\mathbf{z}=\tilde{\mathbf{z}}}\right\|^2_{\ell_2} &\leq \frac{\sigma}{\sqrt{n}} \left\langle \hat{\mathbf{x}}_n(\mathcal{C})|_{\mathbf{z}=\tilde{\mathbf{z}}} - \mathbf{x}^*, \tilde{\mathbf{z}} \right\rangle \\
&= \frac{\sigma}{\sqrt{n}} \left\|\hat{\mathbf{x}}_n(\mathcal{C})|_{\mathbf{z}=\tilde{\mathbf{z}}} - \mathbf{x}^*\right\|_{\ell_2} \left\langle \frac{\hat{\mathbf{x}}_n(\mathcal{C})|_{\mathbf{z}=\tilde{\mathbf{z}}} - \mathbf{x}^*}{\left\|\hat{\mathbf{x}}_n(\mathcal{C})|_{\mathbf{z}=\tilde{\mathbf{z}}} - \mathbf{x}^*\right\|_{\ell_2}}, \tilde{\mathbf{z}} \right\rangle \\
&\leq \frac{\sigma}{\sqrt{n}} \left\|\hat{\mathbf{x}}_n(\mathcal{C})|_{\mathbf{z}=\tilde{\mathbf{z}}} - \mathbf{x}^*\right\|_{\ell_2} \left[\sup_{\mathbf{d} \in T_{\mathcal{C}}(\mathbf{x}^*), \|\mathbf{d}\|_{\ell_2} \leq 1} \langle \mathbf{d}, \tilde{\mathbf{z}} \rangle\right].
\end{aligned}$$

Dividing both sides by $\|\hat{\mathbf{x}}_n(\mathcal{C})|_{\mathbf{z}=\tilde{\mathbf{z}}} - \mathbf{x}^*\|_{\ell_2}$, squaring both sides, and finally taking expectations completes the proof. □

Note that the basic structure of the error bound provided by the estimator [**2**] in fact holds for an *arbitrary* distribution on the noise $\mathbf{z}$ with the Gaussian squared-complexity suitably modified. However, we focus for the rest of this paper on the Gaussian case, $\mathbf{z} \sim \mathcal{N}(0, I_{p \times p})$.

To summarize in words, the mean squared error is bounded by the noise variance times the Gaussian squared-complexity of the

Downloaded from https://www.pnas.org by 103.238.106.114 on December 13, 2022 from IP address 103.238.106.114.

normalized tangent cone with respect to $\mathcal{C}$ at the true parameter $\mathbf{x}^*$. Essentially, it measures the amount of noise restricted to the tangent cone, which is intuitively reasonable because only the noise that moves one away from $\mathbf{x}^*$ in a feasible direction in $\mathcal{C}$ must contribute toward the error. Therefore, if the convex constraint set $\mathcal{C}$ is "sharp" at $\mathbf{x}^*$ so that the cone $T_{\mathcal{C}}(\mathbf{x}^*)$ is "narrow," the error is then small. At the other extreme, if the constraint set $\mathcal{C} = \mathbb{R}^p$, the error is then $\frac{\sigma^2}{n}p$ as one would expect.

Although *Proposition 4* is useful and indeed will suffice for the purposes of demonstrating time-data tradeoffs in the next section, there are a couple of shortcomings in the result as stated. First, suppose the signal set $\mathcal{S}$ is contained in a ball around the origin, with the radius of the ball being small relative to the noise variance $\frac{\sigma^2}{n}$. In such a setting, the estimator $\hat{\mathbf{x}} = 0$ leads to a smaller mean squared error than one would obtain from *Proposition 4*. Second, and somewhat more subtly, suppose that one does not have a perfect bound on the size of the signal set. For concreteness, consider a setting in which $\mathcal{S}$ is a set of sparse vectors with bounded ℓ_1 norm, in which case a good choice for the constraint set $\mathcal{C}$ in the estimator **[1]** is an appropriately scaled ℓ_1 ball. However, if we do not know the ℓ_1 norm of $\mathbf{x}^*$ a priori, we may then end up using a constraint set $\mathcal{C}$ such that $\mathbf{x}^*$ does not belong to $\mathcal{C}$ (hence, $\mathbf{x}^*$ is an infeasible solution) or such that $\mathbf{x}^*$ lies strictly in the interior of $\mathcal{C}$ [hence, $T_{\mathcal{C}}(\mathbf{x}^*)$ is all of $\mathbb{R}^p$]. Both these situations are undesirable because they limit the applicability of *Proposition 4* and provide very loose error bounds. The following result addresses these shortcomings by weakening the assumptions of *Proposition 4*.

Proposition 5.

Let $\mathbf{x}^* \in \mathcal{S} \subset \mathbb{R}^p$, and let $\mathcal{C} \subseteq \mathbb{R}^p$ be a convex set. Suppose there exists a point $\tilde{\mathbf{x}} \in \mathcal{C}$ such that $\mathcal{C} - \tilde{\mathbf{x}} = Q_1 \oplus Q_2$, with Q_1, Q_2 lying in orthogonal subspaces of $\mathbb{R}^p$ and $Q_2 \subseteq \alpha B^p_{\ell_2}$ for $\alpha \geq 0$. We then have that

$$\mathbb{E}\left[\left\|\mathbf{x}^* - \hat{\mathbf{x}}_n(\mathcal{C})\right\|^2_{\ell_2}\right] \leq 6\left[\frac{\sigma^2}{n} g\left(\mathrm{cone}(Q_1) \cap B^p_{\ell_2}\right) + \left\|\mathbf{x}^* - \tilde{\mathbf{x}}\right\|^2_{\ell_2} + \alpha^2\right].$$

Here, $\mathrm{cone}(Q_1)$ is the conic hull of Q_1.

The proof of this result is presented in *SI Appendix*. A number of remarks are in order here. With respect to the first shortcoming in *Proposition 4* stated above, if $\mathcal{C}$ is chosen such that $\mathcal{S} \subset \mathcal{C}$, one can set $\tilde{\mathbf{x}} = \mathbf{x}^*$, $Q_1 = 0$ and $Q_2 = \mathcal{C} - \tilde{\mathbf{x}}$ in *Proposition 5* and can readily obtain a bound that scales only with the diameter of the convex constraint set $\mathcal{C}$. With regard to the second shortcoming in *Proposition 4* described above, if a point $\tilde{\mathbf{x}} \in \mathcal{C}$ near $\mathbf{x}^*$ has a narrow tangent cone $T_{\mathcal{C}}(\tilde{\mathbf{x}})$, one can then provide an error bound with respect to $g(T_{\mathcal{C}}(\tilde{\mathbf{x}}))$ with an extra additive term that depends on $\|\mathbf{x}^* - \tilde{\mathbf{x}}\|^2_{\ell_2}$; this is done by setting $Q_1 = \mathcal{C} - \tilde{\mathbf{x}}$ and $Q_2 = 0$ (thus, $\alpha = 0$) in *Proposition 5*. More generally, *Proposition 5* incorporates both of these improvements in a single error bound with respect to an arbitrary point $\tilde{\mathbf{x}} \in \mathcal{C}$; thus, one can further optimize the error bound over the choice of $\tilde{\mathbf{x}} \in \mathcal{C}$ (as well as the choice of the decomposition Q_1 and Q_2).

Properties and Computation of Gaussian Squared-Complexity. We record some properties of the Gaussian squared-complexity that are subsequently useful when we demonstrate concrete time-data tradeoffs. It is clear that $g(\cdot)$ is monotonic with respect to set nesting [i.e., $g(\mathcal{D}_1) \leq g(\mathcal{D}_2)$ for sets $\mathcal{D}_1 \subseteq \mathcal{D}_2$]. If $\mathcal{D}$ is a subspace, one can then check that $g(\mathcal{D}) = \dim(\mathcal{D})$. To estimate squared-complexities of families of cones, one can imagine appealing to techniques similar to those used for estimating Gaussian complexities of sets (36, 37). Most prominent among these are arguments based on covering number and metric entropy bounds. However, these arguments are frequently not sharp and introduce extraneous log-factors in the resulting error bounds.

In a recent paper by Chandrasekaran et al. (38), sharp upper bounds on the Gaussian complexities of normalized cones have been established for families of cones of interest in a class of linear inverse problems. The (square of the) Gaussian complexity can be upper-bounded by the Gaussian squared-complexity $g(\mathcal{D})$ via Jensen's inequality:

$$\mathbb{E}\left[\sup_{\mathbf{d} \in \mathcal{D}} \langle \mathbf{d}, \mathbf{g} \rangle\right]^2 \leq g(\mathcal{D}),$$

where $\mathbf{g}$ is a standard normal vector. In fact, most of the bounds in the paper by Chandrasekaran et al. (38) were obtained by bounding $g(\mathcal{D})$; thus, they are directly relevant to our setting. In the rest of this section, we present the bounds on $g(\mathcal{D})$ from the paper by Chandrasekaran et al. (38) that will be used in this paper, deferring to that paper for proofs in most cases. In some cases, the proofs do require modifications with respect to their counterparts in the paper by Chandrasekaran et al. (38), and for these cases, we give full proofs in *SI Appendix*.

The first result, proved in (38), is a direct consequence of convex duality and provides a fruitful general technique to compute sharp estimates of Gaussian squared-complexities. Let $\mathrm{dist}(\mathbf{a}, \mathcal{D})$ denote the ℓ_2 distance from a point $\mathbf{a}$ to the set $\mathcal{D}$.

Lemma 1.

Let $\mathcal{K} \subseteq \mathbb{R}^p$ be a convex cone, and let $\mathcal{K}^* \subseteq \mathbb{R}^p$ be its polar. We then have for any $\boldsymbol{a} \in \mathbb{R}^p$ that

$$\sup_{\mathbf{d} \in \mathcal{K} \cap B^p_{\ell_2}} \langle \mathbf{d}, \mathbf{a} \rangle = \mathrm{dist}(\mathbf{a}, \mathcal{K}^*).$$

Therefore, we have the following result as a simple corollary.

Corollary 6.

Let $\mathcal{K} \subseteq \mathbb{R}^p$ be a convex cone, and let $\mathcal{K}^* \subseteq \mathbb{R}^p$ be its polar. For $\mathbf{g} \sim \mathcal{N}(0, I_{p \times p})$, we have that

$$g\left(\mathcal{K} \cap B^p_{\ell_2}\right) = \mathbb{E}\left[\mathrm{dist}(\mathbf{g}, \mathcal{K}^*)^2\right].$$

Based on the duality result of *Lemma 1* and *Corollary 6*, one can compute the following sharp bounds on the Gaussian squared-complexities of tangent cones with respect to the ℓ_1 norm and nuclear norm balls. These are especially relevant when one wishes to estimate sparse signals or low-rank matrices; in these settings, the ℓ_1 and nuclear norm balls serve as useful constraint sets for denoising because the tangent cones with respect to these sets at sparse vectors and at low-rank matrices are particularly narrow. Both of these results, proved in (38), are used when we describe time-data tradeoffs.

Proposition 7.

Let $\mathbf{x} \in \mathbb{R}^p$ be a vector containing s nonzero entries. Let T be the tangent cone at $\mathbf{x}$ with respect to an ℓ_1 norm ball scaled so that $\mathbf{x}$ lies on the boundary of the ball (i.e., a scaling of the unit ℓ_1 norm ball by a factor $\|\mathbf{x}\|_{\ell_1}$). Then,

$$g\left(T \cap B^p_{\ell_2}\right) \leq 2s \log\left(\frac{p}{s}\right) + \frac{5}{4}s.$$

Next, we state a result, proved in (38), for low-rank matrices and the nuclear norm ball.

Proposition 8.

Let $\mathbf{X} \in \mathbb{R}^{m_1 \times m_2}$ be a matrix of rank r. Let T be the tangent cone at $\mathbf{X}$ with respect to a nuclear norm ball scaled so that $\mathbf{X}$ lies on the boundary of the ball (i.e., a scaling of the unit nuclear norm ball by a factor equal to the nuclear norm of $\mathbf{X}$). Then,

$$g\left(T \cap B^{m_1 m_2}_{\ell_2}\right) \leq 3r(m_1 + m_2 - r).$$

Next, we state and prove a result that allows us to estimate Gaussian squared-complexities of general cones. The bound is based on the volume of the dual of the cone of interest, and the proof involves an appeal to Gaussian isoperimetry (39). A similar result on Gaussian complexities of cones (without the square) was proved by Chandrasekaran et al. (38), but that result does not directly imply our statement, and we therefore give a complete self-contained proof in *SI Appendix*. The volume of a cone

Downloaded from https://www.pnas.org by 103.238.106.114 on December 13, 2022 from IP address 103.238.106.114.

is assumed to be normalized (between 0 and 1); thus, we consider the relative fraction of a unit Euclidean sphere that is covered by a cone.

Proposition 9.

Let $\mathcal{K} \subset \mathbb{R}^p$ be a cone such that its polar $\mathcal{K}^* \subset \mathbb{R}^p$ has a normalized volume of $\mu \in \left(\frac{1}{4}\exp\{-p/20\}, \frac{1}{4e^2}\right)$. For $p \geq 12$, we have that

$$g\left(\mathcal{K} \cap B_{\ell_2}^p\right) \leq 20 \log\left(\frac{1}{4\mu}\right).$$

If a cone is narrow, its polar will then be wide, leading to a large value of μ, and hence a small quantity on the right-hand side of the bound. This result leads to bounds on Gaussian squared-complexity in settings in which one can easily obtain estimates of volumes. One setting in which such estimates are easily obtained is the case of tangent cones with respect to vertex transitive polytopes. We recall that a vertex transitive polytope (23) is one in which there exists a symmetry of the polytope for each pair of vertices mapping the two vertices isomorphically to each other. Roughly speaking, all the vertices in such polytopes are the same. Some examples include the cross-polytope (the ℓ_1 norm ball), the simplex [**4**], the hypercube (the ℓ_∞ norm ball), and many polytopes generated by the action of groups (40). We will see many examples of such polytopes in our examples on time-data tradeoffs; thus, we will appeal to the following corollary repeatedly.

Corollary 10.

*Suppose that $\mathcal{P} \in \mathbb{R}^p$ is a vertex transitive polytope with v vertices, and let **x** be a vertex of this polytope. If $4e^2 \leq v \leq 4\ exp\{p/20\}$,*

$$g(T_{\mathcal{P}}(\mathbf{x})) \leq 20 \log(v/4).$$

Proof: The normal cones at the vertices of $\mathcal{P}$ partition $\mathbb{R}^p$. If the polytope is vertex-transitive, the normal cones are then all equivalent to each other (up to orthogonal transformations). Consequently, the (normalized) volume of the normal cone at any vertex is $1/v$. Because the normal cone at a vertex is polar to the tangent cone, we have the desired result from *Proposition 9*. □

Time-Data Tradeoffs

Preliminaries. We now turn our attention to giving examples of time-data tradeoffs in denoising problems via convex relaxation. As described previously, we must set a desired risk to realize a time-data tradeoff; in the examples in the rest of this section, we will fix the desired risk to be equal to 1 independent of the problem dimension p. Thus, these denoising problems belong to the time-data complexity classes $\mathbb{TD}$ ($t(p)$, $n(p)$, 1) for different runtime constraints $t(p)$ and sample budgets $n(p)$. The following corollary gives the number of samples required to obtain a mean squared error of 1 via convex optimization in our denoising setup. (In each of our time-data tradeoff examples, the signal sets $\mathcal{S}$ and the associated convex relaxations are "symmetric" so that no point in $\mathcal{S}$ is distinguished; therefore, without loss of generality, we compute the risk by considering the mean squared error at an arbitrary point in $\mathcal{S}$ rather than taking a supremum over $\mathbf{x}^* \in \mathcal{S}$.)

Corollary 11.

For $\mathbf{x}^* \in \mathcal{S}$ and with $\mathcal{S} \subseteq \mathcal{C}$ for a closed, convex set $\mathcal{C}$, if

$$n \geq \sigma^2 g\left(T_{\mathcal{C}}(\mathbf{x}^*) \cap B_{\ell_2}^p\right),$$

then, $\mathbb{E}[\|\mathbf{x}^* - \hat{\mathbf{x}}_n(\mathcal{C})\|_{\ell_2}^2] \leq 1$.

Proof: The result follows by a rearrangement of the terms in the bound in *Proposition 4*. □

This corollary states that if we have access to a dataset with n samples, we can then use any convex constraint set $\mathcal{C}$ such that the term on the right-hand side in the corollary is smaller than n. Recalling that larger constraint sets $\mathcal{C}$ lead to larger tangent cones $T_{\mathcal{C}}$, we observe that if n is large, one can potentially use very weak (and computationally inexpensive) relaxations and still obtain a risk of 1. This observation, combined with the important point that the hierarchies of convex relaxations described previously are simultaneously ordered both by approximation quality and by computational tractability, allows us to realize a time-data tradeoff by using convex relaxation as an algorithm-weakening mechanism. A simple demonstration is provided in Fig. 2.

We further consider settings with $\sigma^2 = 1$ and in which our signal sets $\mathcal{S} \subseteq \mathbb{R}^p$ consist of elements that have Euclidean norm on the order of $\sqrt{p}$ (measured from the centroid of $\mathcal{S}$). In such regimes, the James–Stein shrinkage estimator (41) offers about the same level of performance as the maximum-likelihood estimator, and both of these are outperformed in statistical risk by nonlinear estimators of the form [**2**] based on convex optimization.

Finally, we briefly remark on the runtimes of our estimators. The runtime for each of the procedures below is calculated by adding the number of operations required to compute the sample mean $\bar{\mathbf{y}}$ and the number of operations required to solve [**2**] to some accuracy. Hence, if the number of samples used is n and if $f_{\mathcal{C}}(p)$ denotes the number of operations required to project $\bar{\mathbf{y}}$ onto $\mathcal{C}$, the total runtime is then $np + f_{\mathcal{C}}(p)$. Thus, the number of samples enters the runtime calculations as just an additive term. As we process larger datasets, the first term in this calculation becomes larger, but this increase is offset by a more substantial decrease in the second term due to the use of a computationally tractable convex relaxation. We note that such a runtime calculation extends to more general inference problems in which one employs estimators of the form [**2**] but with different loss functions in the objective; specifically, the runtime is calculated as above so long as the loss function depends only on some sufficient statistic computed from the data. If the loss function is instead of the form $\sum_{i=1}^n \ell(\mathbf{x}; \mathbf{y}_i)$ and it cannot be summarized via a sufficient statistic of the data $\{\mathbf{y}_i\}_{i=1}^n$, the number of samples then enters the runtime computation in a multiplicative manner as $\theta(n) f_{\mathcal{C}}(p)$ for some function $\theta(\cdot)$.

Example 1: Denoising Signed Matrices. We consider the problem of recovering signed matrices corrupted by noise:

$$\mathcal{S} = \left\{\mathbf{aa}' | \mathbf{a} \in \{-1, +1\}^{\sqrt{p}}\right\}.$$

We have $\mathbf{a} \in \mathbb{R}^{\sqrt{p}}$ so that $\mathcal{S} \subseteq \mathbb{R}^p$. Inferring such signals is of interest in collaborative filtering, where one wishes to approximate matrices as the sum of a small number of rank-one signed matrices (34). Such matrices may represent, for example, the movie preferences of users as in the Netflix problem.

The tightest convex constraint that one could use in this case is $\mathcal{C} = \text{conv}(\mathcal{S})$, which is the cut polytope [**6**]. To obtain a risk of 1

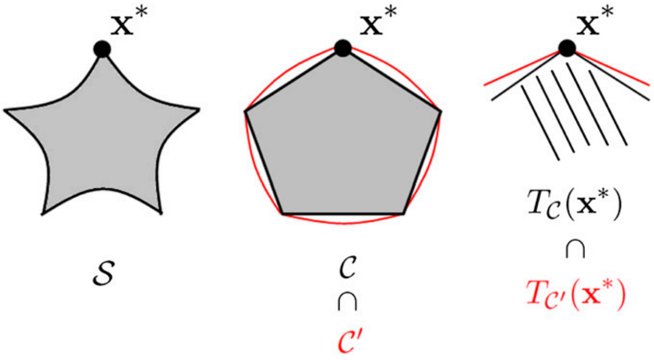

Fig. 2. (*Left*) Signal set $\mathcal{S}$ consisting of $\mathbf{x}^*$. (*Center*) Two convex constraint sets $\mathcal{C}$ and $\mathcal{C}'$, where $\mathcal{C}$ is the convex hull of $\mathcal{S}$ and $\mathcal{C}'$ is a relaxation that is more efficiently computable than $\mathcal{C}$. (*Right*) Tangent cone $T_{\mathcal{C}}(\mathbf{x}^*)$ is contained inside the tangent cone $T_{\mathcal{C}'}(\mathbf{x}^*)$. Consequently, the Gaussian squared-complexity $g(T_{\mathcal{C}}(\mathbf{x}^*) \cap B_{\ell_2}^p)$ is smaller than the complexity $g(T_{\mathcal{C}'}(\mathbf{x}^*) \cap B_{\ell_2}^p)$, so that the estimator $\hat{\mathbf{x}}_n(\mathcal{C})$ requires fewer samples than the estimator $\hat{\mathbf{x}}_n(\mathcal{C}')$ for a risk of at most 1.

with this constraint, one requires $n = c_1\sqrt{p}$ by applying *Corollary 10* and *Corollary 11* based on the symmetry of the cut polytope. The cut polytope is generally intractable to compute. Hence, the best-known algorithms to project onto $\mathcal{C}$ would require a runtime that is superpolynomial in p. Consequently, the total runtime of this algorithm is $c_1p^{1.5}$ + superpoly(p).

A commonly used tractable relaxation of the cut polytope is the elliptope [**5**]. By computing the Gaussian squared-complexity of the tangent cones at rank-one signed matrices with respect to this set, it is possible to show that $n = c_2\sqrt{p}$ leads to a risk of 1 (with $c_2 > c_1$). Furthermore, interior point-based convex optimization algorithms for solving [**2**] that exploit the special structure of the elliptope require $\mathcal{O}(p^{2.25})$ operations (8, 42, 43). (The exponent is a result of the manner in which we define our signal set so that a rank-one signed matrix lives in $\mathbb{R}^p$.) Hence, the total runtime of this procedure is $c_2p^{1.5}$ + $\mathcal{O}(p^{2.25})$.

Finally, an even weaker relaxation of the cut polytope than the elliptope is the unit ball of the nuclear norm scaled by a factor of $\sqrt{p}$; one can verify that the elements of $\mathcal{S}$ lie on the boundary of this set and are, in fact, extreme points. Appealing to *Proposition 8* (using the fact that the elements of $\mathcal{S}$ are rank-one matrices) and *Corollary 11*, we conclude that $n = c_3\sqrt{p}$ samples provide a mean squared error of 1 (with $c_3 > c_2$). Projecting onto the scaled nuclear norm ball can be done by computing a singular value decomposition (SVD) and then truncating the sequence of singular values in descending order when their cumulative sum exceeds $\sqrt{p}$ (in effect, projecting the vector of singular values onto an ℓ_1 ball of size $\sqrt{p}$). This operation requires $\mathcal{O}(p^{1.5})$ operations; thus, the total runtime is $c_3p^{1.5}$ + $\mathcal{O}(p^{1.5})$.

To summarize, the cut-matrix denoising problem lives in the time-data class $\mathbb{TD}$(superpoly(p),$c_1\sqrt{p}$,1), in $\mathbb{TD}(\mathcal{O}(p^{2.25}), c_2\sqrt{p}, 1)$, and in $\mathbb{TD}(\mathcal{O}(p^{1.5}), c_3\sqrt{p}, 1)$, with constants $c_1 < c_2 < c_3$.

Example 2: Ordering Variables. In many data analysis tasks, one is given a collection of variables that are suitably ordered so that the population covariance is banded. Under such a constraint, thresholding the entries of the empirical covariance matrix based on their distance from the diagonal has been shown to be a powerful method for estimation in the high-dimensional setting (44). However, if an ordering of the variables is not known a priori, one must jointly learn an ordering for the variables and estimate their underlying covariance. As a stylized version of this variable ordering problem, let $M \in \mathbb{R}^{\sqrt{p}\times\sqrt{p}}$ be a known tridiagonal matrix (with Euclidean norm $\mathcal{O}(\sqrt{p})$) and consider the following signal set:

$$\mathcal{S} = \left\{\Pi M \Pi' | \Pi \text{ is a } \sqrt{p} \times \sqrt{p} \text{ permutation matrix}\right\}.$$

The matrix M here is to be viewed as a covariance matrix. Thus, the corresponding denoising problem [**1**] is that we wish to estimate a covariance matrix in the absence of knowledge of the ordering of the underlying variables. In a real-world scenario, one might wish to consider covariance matrices M that belong to some class of banded matrices and then construct $\mathcal{S}$ as done here, but we stick with the case of a fixed M for simplicity. Furthermore, the noise in a practical setting is better modeled as coming from a Wishart distribution; again, we focus on the Gaussian case for simplicity.

The tightest convex constraint set that one could use in this case is the convex hull of $\mathcal{S}$, which is generally intractable to compute for arbitrary matrices M. For example, if one were able to compute this set in polynomial time for any tridiagonal matrix M, one would be able to solve the intractable longest path problem (45) (finding the longest path between any two vertices in a graph) in polynomial time. With this convex constraint set, we find using *Corollary 10* and *Corollary 11* that $n = c_1\sqrt{p}\log(p)$ samples would lead to a risk of 1. This follows from the fact that conv($\mathcal{S}$) is a vertex-transitive polytope with about $(\sqrt{p})!$ vertices. Thus, the total runtime is $c_1p^{1.5}log(p)$ + superpoly(p).

An efficiently computable relaxation of conv($\mathcal{S}$) is a scaled ℓ_1 ball (scaled by the ℓ_1 norm of M). Appealing to *Proposition 7* on tangent cones with respect to the ℓ_1 ball and to *Corollary 11*, we find that $n = c_2\sqrt{p}\log(p)$ samples suffice to provide a risk of 1. In applying *Proposition 7*, we note that M is assumed to be tridiagonal, and therefore has $\mathcal{O}(\sqrt{p})$ nonzero entries. The runtime of this procedure is $c_2p^{1.5}$ log(p) + $\mathcal{O}(p$ log(p)).

Thus, the variable ordering denoising problem belongs to $\mathbb{TD}$(superpoly(p),$c_1\sqrt{p}\log(p)$,1) and to $\mathbb{TD}(\mathcal{O}(p^{1.5}\log(p)), c_2\sqrt{p}\log(p), 1)$, with constants $c_1 < c_2$.

Example 3: Sparse PCA and Network Activity Identification. As our third example, we consider sparse PCA (18) in which one wishes to learn from samples a sparse eigenvector that contains most of the energy of a covariance matrix. As a simplified version of this problem, one can imagine a matrix $M \in \mathbb{R}^{\sqrt{p}\times\sqrt{p}}$ with entries equal to $\sqrt{p}/k$ in the top-left $k \times k$ block and zeros elsewhere (so that the Euclidean norm of M is $\sqrt{p}$), and with $\mathcal{S}$ defined as:

$$\mathcal{S} = \left\{\Pi M \Pi' | \Pi \text{ is a } \sqrt{p} \times \sqrt{p} \text{ permutation matrix}\right\}.$$

In addition to sparse PCA, such signal sets are of interest in identifying activity in noisy networks (46) as well as in related combinatorial optimization problems, such as the planted clique problem (45). In the sparse PCA context, Amini and Wainwright (16) study time-data tradeoffs by investigating the sample complexities of two procedures: a simple one based on thresholding and a more sophisticated one based on semidefinite programming. Kolar et al. (46) investigate the sample complexities of a number of procedures ranging from a combinatorial search method to thresholding and sparse SVD. We note that the time-data tradeoffs studied in these two papers (16, 46) relate to the problem of learning the support of the leading sparse eigenvector; in contrast, in our setup, the objective is simply to denoise an element of $\mathcal{S}$. Furthermore, although the Gaussian noise setting is of interest in some of these domains, in a more realistic sparse PCA problem (e.g., the one considered in ref. 16), the noise is Wishart rather than Gaussian as considered here. Nevertheless, we stick with our stylized problem setting because it provides some useful insights on time-data tradeoffs. Finally, the size of the block $k \in \{1, \ldots, \sqrt{p}\}$ depends on the application of interest, and it is typically far from the extremes 1 and $\sqrt{p}$. We will consider the case $k \sim p^{1/4}$ for concreteness. [This setting is an interesting threshold case in the planted clique context (47–49), where $k = p^{1/4}$ is the square root of the number of nodes of the graph represented by M (viewed as an adjacency matrix).]

As usual, the tightest convex constraint set one can use in this setting is the convex hull of $\mathcal{S}$, which is generally intractable to compute; an efficient characterization of this polytope would lead to an efficient solution of the intractable planted-clique problem (finding a fully connected subgraph inside a larger graph). Using this convex constraint set gives an estimator that requires about $n = \mathcal{O}(p^{1/4}\log(p))$ samples to produce a risk-1 estimate. We obtain this threshold by appealing to *Corollary 10* and to *Corollary 11*, as well as to the observation that conv($\mathcal{S}$) is a vertex-transitive polytope with about $\binom{\sqrt{p}}{p^{1/4}}$ vertices. Thus, the overall runtime is $\mathcal{O}(p^{5/4}$ *log*(p)) + superpoly(p).

A convex relaxation of conv($\mathcal{S}$) is the nuclear norm ball scaled by a factor of $\sqrt{p}$ so that the elements of $\mathcal{S}$ lie on the boundary. From *Proposition 8* (observing that the elements of $\mathcal{S}$ are rank-one matrices) and *Corollary 11*, we have that $n = c\sqrt{p}$ samples give a risk-1 estimate with this procedure. As computed in the example with cut matrices, the overall runtime of this nuclear norm procedure is $cp^{1.5}$ + $\mathcal{O}(p^{1.5})$.

In conclusion, the denoising version of sparse PCA lies in $\mathbb{TD}$(superpoly(p), $\mathcal{O}(p^{1/4}$ *log*(p)),1) and in $\mathbb{TD}(\mathcal{O}(p^{1.5}), \mathcal{O}(\sqrt{p}), 1)$.

Example 4: Estimating Matchings. As our final example, we consider signals that represent the set of all perfect matchings in the complete graph. A matching is any subset of edges of a graph such that no node of the graph is incident to more than one edge in the subset, and a perfect matching is a subset of edges in which every node is incident to exactly one edge in the subset. Graph matchings arise in a range of inference problems, such as in chemical structure

Downloaded from https://www.pnas.org by 103.238.106.114 on December 13, 2022 from IP address 103.238.106.114.

Downloaded from https://www.pnas.org by 103.238.106.114 on December 13, 2022 from IP address 103.238.106.114.

analysis (50) and in network monitoring (51). Letting M be the adjacency matrix of some perfect matching in the complete graph on $\sqrt{p}$ nodes, our signal set in this case is defined as follows:

$$\mathcal{S} = p^{1/4}\left\{\Pi M \Pi' | \Pi \text{ is a } \sqrt{p} \times \sqrt{p} \text{ permutation matrix}\right\}.$$

The scaling of $p^{1/4}$ ensures that the elements of $\mathcal{S}$ have Euclidean norm of $\sqrt{p}$. Note that $\mathcal{S} \subset \mathbb{R}^p$. The number of elements in $\mathcal{S}$ is $\frac{(\sqrt{p})!}{\left(\frac{\sqrt{p}}{2}\right)!\ 2^{\sqrt{p}/2}}$ when $\sqrt{p}$ is an even number (this number is obtained by computing the product of all the odd integers up to $\sqrt{p}$).

The tightest convex relaxation in this case is the convex hull of $\mathcal{S}$. Unlike the previous three cases, projecting onto this convex set is, in fact, a polynomial-time operation, with a runtime of about $\mathcal{O}(p^5)$. [Edmonds' blossom algorithm (52) for computing maximum-weight matchings in polynomial time leads to a separation oracle for this perfect matching polytope. Subsequently, Padberg and Rao (53) developed a faster separation oracle for the perfect matching polytope. These separation oracles, in turn, lead to polynomial-time projection algorithms via the ellipsoid method (42).] Appealing to *Corollary 10*, to *Corollary 11*, and to the fact that conv($\mathcal{S}$) is a vertex-transitive polytope, we have that $n = c_1\sqrt{p}\log(p)$ samples provides a risk-1 estimate. Hence the overall runtime is $c_1 p^{1.5} \log(p) + \mathcal{O}(p^5)$.

A tractable relaxation of the perfect matching polytope is a hypersimplex (23), obtained by taking the convex hull of all $\sqrt{p}\times\sqrt{p}$ matrices consisting of $\sqrt{p}$ ones and the other entries being equal to zero. We scale this hypersimplex by a factor of $p^{1/4}$ so that the elements of $\mathcal{S}$ are on the boundary. The hypersimplex is also a vertex-transitive polytope, like the perfect matching polytope, but with about $\binom{p}{\sqrt{p}}$ entries. Hence, from *Corollary 10* and *Corollary 11*, we have that $n = c_2\sqrt{p}\log(p)$ samples will provide a risk-1 estimate. Furthermore, projecting onto the hypersimplex is a very efficient operation based on sorting, and it has a runtime of $\mathcal{O}(p \log(p))$. Consequently, the total runtime of this procedure is $c_2 p^{1.5}\log(p) + \mathcal{O}(p \log(p))$.

In summary, the matching estimation problem is a member of $\mathbb{TD}(\mathcal{O}(p^5), c_1\sqrt{p}\log(p), 1)$ and of $\mathbb{TD}(\mathcal{O}(p^{1.5}\log(p)), c_2\sqrt{p}\log(p), 1)$ with constants $c_1 < c_2$.

Some Observations. A curious observation that we may take away from these examples is that it is possible to obtain substantial speedups computationally with just a constant factor increase in the size of the dataset. This suggests that in settings in which obtaining additional data is inexpensive, it may be more economical to procure more data and use a more basic computational infrastructure rather than to process limited data using powerful and expensive computers.

Our second observation is relevant to all the examples above, but we highlight it in the context of denoising cut matrices. In that setting, one can use an even weaker relaxation of the cut polytope than the nuclear norm ball, such as the Euclidean ball (suitably scaled). Although projection onto this set is extremely efficient [requiring $\mathcal{O}(p)$ operations as opposed to $\mathcal{O}(p^{1.5})$ operations for projecting onto the nuclear norm ball], the number of samples required to achieve a risk of 1 with this approach is $\mathcal{O}(p)$: Computing the sample mean with so many samples requires $\mathcal{O}(p^2)$ operations, which leads to an overall runtime that is greater than the runtime $\mathcal{O}(p^{1.5})$ for the nuclear norm approach. This point highlights an important tradeoff: If our choice of algorithms is between nuclear norm projection and Euclidean projection, and if we are, in fact, given access to $\mathcal{O}(p)$ data samples, it makes sense computationally to retain only $\mathcal{O}(\sqrt{p})$ samples for the nuclear norm procedure and to throw away the remaining data. This provides a concrete illustration of several key issues. Aggregating massive datasets can frequently be very expensive computationally (relative to the other subsequent processing), and the number of operations required for this step must be taken into account. (Note that the aggregation step is more time-consuming than the subsequent projection step in the ℓ_1-ball projection procedure for ordering variables and in the hypersimplex projection method for denoising matchings.) Consequently, in some cases, it may make sense to throw away some data if preprocessing the full massive dataset is time-consuming. Hence, one may not be able to avail oneself of weaker postaggregation algorithms if these methods require such a large amount of data to achieve a desired risk that the aggregation step is expensive. This point goes back to the floor in Fig. 1 in which one imagines a cutoff in the number of samples beyond which more data are not helpful in reducing computational runtime. Such a threshold, of course, depends on the space of algorithms one employs, and in the cut polytope context with the particular algorithms considered here, the threshold occurs at $\mathcal{O}(\sqrt{p})$ samples. Our discussion is premised on the point that both the data aggregation step and the subsequent projection step use the same computational infrastructure. In practice, however, the data aggregation step may be effectively parallelized, and the runtime calculations associated with such modifications will result in different time-data tradeoffs and data floors than those described above.

Conclusions

In this paper, we considered the problem of reducing the computational complexity of an inference task as one has access to larger datasets. The traditional goal in the theory of statistical inference is to understand the tradeoff in an estimation problem between the amount of data available and the risk attainable via some class of procedures. In an age of plentiful data in many settings and computational resources being the principal bottleneck, we believe that an increasingly important objective is to investigate the tradeoffs between computational and sample complexities. As one pursues this line of thinking, it becomes clear that a central theme must be the ability to weaken an inference procedure as one has access to larger datasets. Accordingly, we proposed convex relaxation as an algorithm-weakening mechanism, and we investigated its efficacy in a class of denoising tasks. Our results suggest that such methods are especially effective in achieving time-data tradeoffs in high-dimensional parameter estimation.

We close our discussion by outlining some exciting future research directions. Because algorithm weakening is central to the viewpoint described in this paper, it should come as no surprise that several of the directions listed below involve interaction with important themes in computer science.

Computation with Streaming or Inhomogeneous Data. In some massive data problems, one is presented with a stream of input data rather than a large fixed dataset, and an estimate may be desired after a fixed amount of time independent of the rate of the input stream. In other domains, the data may be heterogeneous or may be generated from a nonstationary source, as opposed to the stationary case considered in this paper. In these settings, alternative viewpoints on tradeoffs to the one presented in this paper might be more appropriate. For example, with streaming data, one could keep the runtime fixed and trade-off the risk with the rate of the input stream. One can imagine algorithm-weakening mechanisms, dependent on the rate of the data stream, in which the initial data points are processed using sophisticated algorithms and subsequent samples are processed more coarsely. Understanding the tradeoffs in such settings with streaming or inhomogeneous data is of interest in a range of applications.

Alternative Algorithm-Weakening Mechanisms. The notion of weakening an inference algorithm is key to realizing a time-data tradeoff. Although convex relaxation methods provide a powerful and general approach, a number of other weakening mechanisms are potentially relevant. For example, processing data more coarsely by quantization, dimension reduction, and clustering may be natural in some contexts. Coresets, which originated in the computational geometry community, summarize a large set of points via a small collection (e.g., ref. 54 and the references therein), and they could also provide a powerful algorithm-weakening

Downloaded from https://www.pnas.org by 103.238.106.114 on December 13, 2022 from IP address 103.238.106.114.

mechanism. Finally, we would like to mention a computer hardware concept that has implications for massive data analysis. Recent approaches to designing computer chips are premised on the idea that many tasks do not require extremely accurate computation. If one is willing to tolerate small, random errors in arithmetic computations (e.g., addition, multiplication), it may be possible to design chips that consume less power and are faster than traditional, more accurate chips. Translated to a data analysis context, such design principles may provide a hardware-based algorithm-weakening mechanism.

Measuring Quality of Approximation of Convex Sets. In the mathematical optimization and theoretical computer science communities, relaxations of convex sets have provided a powerful toolbox for designing approximation algorithms for intractable problems, most notably those arising in combinatorial optimization. The manner in which the quality of a relaxation translates to the quality of an approximation algorithm is usually quantified based on the *integrality gap* between the original convex set and its approximation (4). However, the quantity of interest in a statistical inference context in characterizing the quality of approximations is based on ratios of Gaussian squared-complexities of tangent cones. These two quantifications can be radically different; indeed, several of the relaxations presented in our time-data tradeoff examples that are useful in an inferential setting would provide poor performance in a combinatorial optimization context. More broadly, those examples demonstrate that weak relaxations frequently provide as good estimation performance as tighter ones with just an increase of a constant factor in the number of data samples. This observation suggests a potentially deeper result along the following lines. Many computationally intractable convex sets for which there exist no tight efficiently computable approximations as measured by integrality gap can nonetheless be well-approximated by computationally tractable convex sets if the quality of approximation is measured based on statistical inference objectives.

ACKNOWLEDGMENTS. We thank Pablo Parrilo, Benjamin Recht, and Parikshit Shah for many insightful conversations. We also thank Alekh Agarwal, Peter Bühlmann, Emmanuel Candès, Robert Nowak, James Saunderson, Leonard Schulman, and Martin Wainwright for helpful questions and discussions. This material is based on work supported, in part, by the US Army Research Laboratory and the US Army Research Office under Contract/Grant W911NF-11-1-0391.

1. Lai TL (2001) Sequential analysis: Some classical problems and new challenges. *Statistica Sinica* 11(2):303–408.
2. Arora S, Barak B (2009) *Computational Complexity: A Modern Approach* (Cambridge Univ Press, New York).
3. Bühlmann P, van de Geer S (2011) *Statistics for High-Dimensional Data: Methods, Theory and Applications* (Springer, Berlin).
4. Vazirani V (2004) *Approximation Algorithms* (Springer, Berlin).
5. Johnstone IM (2011) Gaussian estimation: Sequence and wavelet models. Available at www-stat.stanford.edu/~imj/. Last accessed February 26, 2013.
6. Donoho DL (1995) Denoising by soft thresholding. *IEEE Trans Inf Theory* 41:613–627.
7. Donoho DL, Johnstone IM (1998) Minimax estimation via wavelet shrinkage. *Annals of Statistics* 26(3):879–921.
8. Boyd SP, Vandenberghe L (2004) *Convex Optimization* (Cambridge Univ Press, New York).
9. Shalev-Shwartz S, Shamir O, Tromer E (2012) Using more data to speed up training time. *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics*, April 21–23, 2012, La Palma, Canary Islands, pp 1019–1027.
10. Decatur S, Goldreich O, Ron D (1998) Computational sample complexity. *SIAM J Sci Comput* 29:854–879.
11. Servedio R (2000) Computational sample complexity and attribute-efficient learning. *Journal of Computer and Systems Sciences* 60(1):161–178.
12. Birnbaum A, Shalev-Shwartz S (2012) Learning halfspaces with the zero-one loss: time-accuracy tradeoffs. *Advances in Neural Information Processing Systems 25*, eds Bartlett P, Pereira FCN, Burges CJC, Bottou L, Weinberger KQ, pp 935–943.
13. Bottou L, Bousquet O (2008) The tradeoffs of large scale learning. *Advances in Neural Information Processing Systems 20*, eds Platt JC, Koller D, Singer Y, Roweis S (MIT Press, Cambridge, MA), pp 161–168.
14. Shalev-Schwartz S, Srebro N (2008) SVM optimization: Inverse dependence on training set size. *Proceedings of the 25th Annual International Conference on Machine Learning*, eds McCallum A, Roweis S (Omnipress, Helsinki, Finland), pp 928–935.
15. Perkins TJ, Hallett MT (2010) A trade-off between sample complexity and computational complexity in learning Boolean networks from time-series data. *IEEE/ ACM Trans Comput Biol Bioinformatics* 7(1):118–125.
16. Amini A, Wainwright M (2009) High-dimensional analysis of semidefinite programming relaxations for sparse principal component analysis. *Ann Stat* 37:2877–2921.
17. Kolda T (2001) Orthogonal tensor decompositions. *SIAM Journal on Matrix Analysis* 23:243–255.
18. Johnstone IM, Lu AY (2009) On consistency and sparsity for principal components analysis in high dimensions. *J Am Stat Assoc* 104(486):682–693.
19. Agarwal A, Bartlett P, Duchi J (2012) Oracle inequalities for computationally adaptive model selection. *arXiv*:1208.0129.
20. Nesterov Y, Nemirovskii A (1995) *Interior-Point Polynomial Algorithms in Convex Programming* (Society for Industrial and Applied Mathematics, Philadelphia).
21. Candès EJ, Romberg J, Tao T (2006) Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Trans Inf Theory* 52:489–509.
22. Donoho DL (2006) Compressed sensing. *IEEE Trans Inf Theory* 52:1289–1306.
23. Ziegler G (1995) *Lectures on Polytopes* (Springer, Berlin).
24. Goemans M (2012) *Smallest Compact Formulation for the Permutahedron. Technical Report* (Department of Mathematics, Massachusetts Institute of Technology, Cambridge, MA).
25. Gouveia J, Parrilo P, Thomas R (2012) Lifts of convex sets and cone factorizations. *Mathematics of Operations Research*, 10.1287/moor.1120.0575.
26. Goemans M, Williamson D (1995) Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM* 42(6):1115–1145.
27. Sherali HD, Adams WP (1990) A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems. *SIAM Journal on Discrete Mathematics* 3(3):411–430.
28. Bochnak J, Coste M, Roy M (1988) *Real Algebraic Geometry* (Springer, Berlin).
29. Parrilo PA (2000) Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization. PhD thesis (California Institute of Technology, Pasadena, CA).
30. Parrilo PA (2003) Semidefinite programming relaxations for semialgebraic problems. *Mathematical Programming* 96(2):293–320.
31. Lasserre JB (2001) Global optimization with polynomials and the problem of moments. *SIAM J Optim* 11:796–817.
32. Gouveia J, Parrilo P, Thomas R (2010) Theta bodies for polynomial ideals. *SIAM J Optim* 20:2097–2118.
33. Renegar J (2006) Hyperbolic programs and their derivative relaxations. *Foundations of Computational Mathematics* 6(1):59–79.
34. Srebro N, Shraibman A (2005) *Learning Theory*. 18th Annual Conference on Learning Theory, June 27–30, 2005, Bertinoro, Italy. Lecture Notes in Computer Science 3559, eds Auer P, Meir R (Springer).
35. Rockafellar RT (1970) *Convex Analysis* (Princeton Univ Press, Princeton).
36. Dudley RM (1967) The sizes of compact subsets of Hilbert space and continuity of Gaussian processes. *Journal of Functional Analysis* 1(3):290–330.
37. Bartlett P, Mendelson S (2002) Rademacher and Gaussian complexities: risk bounds and structural results. *J Mach Learn Res* 3:463–482.
38. Chandrasekaran V, Recht B, Parrilo P, Willsky A (2012) The convex geometry of linear inverse problems. *Foundations of Computational Mathematics* 12(6):805–849.
39. Ledoux M (2000) *The Concentration of Measure Phenomenon* (American Mathematical Society, Providence, RI).
40. Sanyal R, Sottile F, Sturmfels B (2011) Orbitopes. *Mathematika* 57:275–314.
41. James W, Stein C (1961) Estimation with quadratic loss. *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability*, ed Neyman J (University of California Press, Berkeley, CA), pp 361–379.
42. Ben Tal A, Nemirovskii A (2001) *Lectures on Modern Convex Optimization* (Society for Industrial and Applied Mathematics, Philadelphia).
43. Higham N (2002) Computing the nearest correlation matrix—A problem from finance. *IMA Journal of Numerical Analysis* 22(3):329–343.
44. Bickel P, Levina L (2008) Regularized estimation of large covariance matrices. *Annals of Statistics* 36(1):199–227.
45. Garey M, Johnson D (1979) *Computers and Intractability: A Guide to the Theory of NP-Completeness* (Freeman, New York).
46. Kolar M, Balakrishnan S, Rinaldo A, Singh A (2011) Minimax localization of structural information in large noisy matrices. *Neural Information Processing Systems 24*, eds Shawe-Taylor J, Zemel RS, Bartlett P, Pereira, FCN, Weinberger, pp 909–917.
47. Alon N, Krivelevich M, Sudakov B (1998) Finding a large hidden clique in a random graph. *Random Structures Algorithms* 13:457–466.
48. Feige U, Krauthgamer R (2000) Finding and certifying a large hidden clique in a semirandom graph. *Random Structures Algorithms* 16:195–208.
49. Ames BPW, Vavasis SA (2011) Nuclear norm minimization for the planted clique and biclique problems. *Mathematical Programming, Series B* 129:69–89.
50. Rouvray DH, Balaban AT (1979) Chemical applications of graph theory. *Applications of Graph Theory* (Academic Press, London), pp 177–221.
51. Shoubridge P, Krarne M, Ray D (1999) Detection of abnormal change in dynamic networks. *Proceedings of Information, Decision, and Control,* pp 557–562.
52. Edmonds J (1965) Maximum matching and a polyhedron with 0-1 vertices. *J Res Natl Bur Stand* 69B:125–130.
53. Padberg M, Rao M (1982) Odd minimum cut-sets and b-matchings. *Mathematics of Operations Research* 7(1):67–80.
54. Feldman D, Langberg M (2011) A unified framework for approximating and clustering data. *Proceedings of the Symposium on the Theory of Computing,* pp 569–578.