

---

# Worst-Case Analysis for Randomly Collected Data

---

**Justin Y. Chen**  
MIT  
justc@mit.edu

**Gregory Valiant**  
Stanford University  
gvaliant@cs.stanford.edu

**Paul Valiant**  
IAS and Purdue University  
pvaliant@gmail.com

## Abstract

We introduce a framework for statistical estimation that leverages knowledge of how samples are collected but makes no distributional assumptions on the data values. Specifically, we consider a population of elements  $\{1, \dots, n\}$  with corresponding values  $x_1, \dots, x_n$ . We observe the values for a *sample* set  $A \subset \{1, \dots, n\}$  and wish to estimate some statistic of the values for a *target* set  $B \subset \{1, \dots, n\}$  where  $B$  could be the entire set. Crucially, we assume that the sets  $A$  and  $B$  are drawn according to some known joint distribution  $(A, B) \sim P$  over pairs of subsets of  $\{1, \dots, n\}$ . A given estimation algorithm is evaluated based on its *worst-case, expected error* where the expectation is with respect to the distribution  $P$  from which the sample  $A$  and target set  $B$  are drawn, and the worst-case is with respect to the data values  $x_1, \dots, x_n$ . Within this general framework we give an efficient algorithm to find an estimator for the target mean, as a weighted combination of the input sample—where the weights are a function of the distribution  $P$  and the identities of the elements in the sample and target sets  $A, B$ . We show that the worst-case expected error achieved by this estimator is at most a multiplicative  $\pi/2$  factor worse than the optimum for such estimators. A component of this algorithm can also be used to approximate the worst-case expected error of a given estimator. The algorithm and proof leverage a surprising connection to the Grothendieck problem. We extend these results to the setting of linear regression, where each datapoint is not a scalar but a labeled vector  $(x_i, y_i) \in \mathbb{R}^{d+1}$ . Our framework, which makes no distributional assumptions on the data values but rather relies on knowledge of the data collection process via the distribution  $P$ , is a significant departure from the typical statistical estimation framework and introduces a uniform algorithmic analysis for the many natural settings where membership in a sample may be correlated with data values, such as when probabilities of sampling vary as in “importance sampling”, when individuals are recruited into a sample via a social network as in “snowball sampling” or “respondent-driven sampling” [12, 14] or when samples have chronological structure as in “selective prediction” [10, 21]. We experimentally demonstrate the benefit of this framework and our algorithm in comparison to standard estimators, for several such settings.

## 1 Modeling Data Collection

For many real-world estimation or prediction problems, it is not unreasonable to assume that data values are drawn independently from some underlying distribution. Correspondingly, there is an enormous body of work developing algorithms suited for such settings as well as for related settings with slightly weaker assumptions such as exchangeability or assumptions found in various models in robust statistics or robust learning. By contrast, there are also many settings in which we know very little about the underlying data values, and any sort of distributional assumption would be problematic. For such settings, however, we might have some knowledge or control over the process by which data is collected. How can we design algorithms for estimating basic statistics that are optimal for a

given data collection process? For which data collection processes is accurate estimation or learning possible, *even for worst-case data*? Surprisingly, there seems to be little work on such questions.

We introduce a general framework in which to study these questions. Consider a set of  $n$  indices  $\{1, \dots, n\}$  with corresponding values  $x = \{x_1, \dots, x_n\}$ , and a distribution  $P$  over pairs of subsets  $(A, B)$  where  $A, B \subset \{1, \dots, n\}$ . We call  $A$  the *sample* set and  $B$  the *target* set. At inference time, a pair is drawn  $(A, B) \sim P$  and we are given access to the data values in  $x$  indexed by set  $A$ , namely  $x_A$ . Our task is to use  $A, B$ , and  $x_A$  to estimate some statistic of the values indexed by set  $B$ ,  $\sigma(x_B)$ . For example, if  $\sigma(\cdot)$  is the arithmetic mean and  $B = \{1, \dots, n\}$ , then our goal is to use the sample to estimate the population mean.

Given an estimator, we consider the *worst-case, expected error*. The worst-case is with respect to the data values  $\{x_i\}$ , and the expectation is with respect to the distribution over subsets  $P$ . This model corresponds to the setting where we have knowledge of the sample collection process, but do not make distributional assumptions on the data values. Throughout, we will be interested in both the question of evaluating the worst-case expected error of a given estimator, as well as the more challenging task of computing the best estimator for a given  $P$ —namely the estimator that minimizes this worst-case expected error under the sampling distribution  $P$ .

Below, we illustrate how this framework captures common data collection processes including processes in which certain individuals are biased towards or against being sampled, processes with dependencies such as "snowball sampling" or "chain sampling" where membership in the sample is governed by stochastic processes (e.g. over a social network), and settings where samples must satisfy chronological constraints (e.g. one uses a sample from the past to make predictions about a target in the future). We begin with a standard example.

**Example 1** (Independent Samples). *Consider the setting where the distribution  $P$  corresponds to including each element  $i \in \{1, \dots, n\}$  in the sample set  $A$  independently with probability  $p$ , and the goal is to estimate the mean of the target set  $B = \{1, \dots, n\}$ . In this setting, provided the data values  $x_1, \dots, x_n$  are bounded (or have bounded variance in the sense that  $\frac{1}{n} \sum_i x_i^2 = O(1)$ ), then the sample mean will concentrate, and even for worst-case datasets, the estimator that simply returns the sample mean will have expected squared error  $O(\frac{1}{pn})$ . In the related setting where each  $i$  is included in the sample independently, but with a possibly distinct probability  $p_i$ , the worst-case expected error framework corresponds to not making assumptions on how the data values  $x_i$  vary with the sampling probabilities  $p_i$ . An estimator for this setting with small worst-case expected error will have good performance even if the  $x_i$ 's are, for example, correlated with the  $p_i$ 's in a pernicious way.*

The example above illustrates one way in which accurate estimation is possible, even for worst-case data: namely if the sample mean (and target mean) concentrates. The following example illustrates that accurate prediction can still be possible, even for a distribution for which the target and sample means have *constant* variance with respect to the randomness in  $(A, B) \sim P$ . This example captures a setting where the distribution,  $P$ , respects *chronological* constraints, in the sense that, for any target/sample sets that have non-zero probability under  $P$ , if  $i$  is in the sample set and  $j$  is in the target set,  $i < j$ . Such constraints mirror the many settings where the sample set corresponds to past data, and the target set corresponds to data that will be received in the future. Here, the worst-case component of our framework corresponds to not making assumptions that the world is "stationary"—future elements might not be like past elements.

**Example 2** ("Selective Prediction"). *Consider the joint sample/target distribution  $P$  corresponding to the following process: a time  $t$  is drawn uniformly from  $\{1, \dots, n-1\}$ , and the sample set is  $\{1, \dots, t\}$ . Then  $w$  is drawn uniformly from  $\{1, 2, 4, 8, \dots, 2^{\log n-1}\}$  and the target set is  $\{t+1, \dots, \min(t+w, n)\}$ . This prediction task corresponds to choosing a day at random, and deciding to make a prediction about the average change in the stock market over the next  $w$  days. In this setting, the main results in [10, 21] imply that if the data values are all bounded, then there exists an estimator for the target mean whose worst-case expected squared error is  $O(1/\log n)$ , and that this is optimal to constant factors. The prediction algorithm achieving this performance is extremely simple: when asked to predict the mean of the next  $w$  data items after  $t$ , return the mean of the most recent  $w$  sample values,  $\frac{1}{w} \sum_{i=t-w+1}^t x_i$ . The surprising aspect of this example is that subconstant expected error is achievable, despite there being no distributional assumptions on the values and hence no guarantees that future data are like past data. The randomness in both time  $t$  and the window length  $w$  of the target set (which define the distribution  $P$ ) are both essential for achieving*

*subconstant worst-case expected accuracy: if either  $t$  or  $w$  is any fixed value, then the worst-case estimation error becomes constant.*

A third example that fits cleanly within our framework is the class of data collection schemes referred to as *snowball sampling*, *respondent driven sampling* or *chain sampling* [12, 14]. In such a scheme, people who have contributed data are asked (or incentivized) to recruit their acquaintances to contribute data, and the pool of respondents grows, like a snowball. These schemes are frequently used to collect data from sensitive populations, such as drug users. Our framework provides a natural way to design and evaluate estimators for data collected via such sampling processes:

**Example 3** (Snowball Sampling). *Suppose elements  $\{1, \dots, n\}$  are located at nodes of a social network. A sample is drawn by independently selecting one (or several) indices; each element in the sample then repeatedly “recruits” each of its friends in the social network (say independently with probability  $p$ ). The sample will then correspond to the elements that have been recruited in the first  $t$  iterations of this “viral” process. The target set could correspond to those nodes recruited in iterations  $t + 1, \dots, t + w$  for some horizon  $w$ , or the target set could be the entire population,  $\{1, \dots, n\}$ . How do structural properties of the underlying social network translate into positive or negative results on the worst-case expected performance of standard estimators, or an optimal estimator? And how does such an optimal estimator leverage knowledge of the network structure? While we do not have simple rules-of-thumb for these questions, our main results apply to such snowball sampling processes, and we evaluate our algorithm empirically in such a setting in Section 3.*

While the notion of worst-case expected error is in terms of a sampling distribution,  $P$ , one of the main motivations for this framework is to guide the choice of estimator for the many real-world settings in which the actual sampling distribution is *unknown*. In such settings, one could construct various different plausible  $P$ ’s—for example capturing different hypothetical types of correlation in inclusion in the sample, different response rates for different hypothetical demographic groups, etc. Using the worst-case expected error framework, one could then rigorously evaluate the stability of potential estimators with respect to these various “plausible”  $P$ ’s. This may prove to be a useful alternative to the more standard approach of evaluating estimators with respect to different assumptions on the data values.

## 1.1 Summary of Results

Our main results are efficient algorithms for approximately computing the worst-case expected error of a given estimator with respect to a given distribution  $P$  (or sample access to  $P$ ), and for finding an estimator that approximately minimizes this error with respect to  $P$ . We state these results in the setting where the statistic of interest is the *mean* of the target set. Such results immediately extend to yield analogous statements for any statistic that can be expressed as the average of some functional of each data point (e.g. the variance, or higher moments). We further apply these results in a black-box fashion to yield results in the setting where each datapoint is a labeled vector  $(x_i, y_i)$ , and the “statistic” of interest is the optimal linear regression model  $\beta$  for the target set, so that  $y_i \approx x_i^\top \beta$ .

Our results in the mean estimation setting focus on *semilinear* schemes, in which estimates can be expressed as a linear combination of the sample values, where the linear coefficients can depend arbitrarily on the indices (but not values) of the sample and target sets  $A, B$  and the distribution  $P$ .

**Definition 1.** *A semilinear estimation algorithm,  $L$ , is a mapping from a set of sample elements  $A = (\alpha_1, \dots, \alpha_{|A|}) \subset \{1, \dots, n\}$  and set of target elements  $B = (\beta_1, \dots, \beta_{|B|}) \subset \{1, \dots, n\}$  to a list of  $|A|$  weights,  $w_1, \dots, w_{|A|}$  that may depend on  $A, B$ , and  $P$ . The estimate produced by  $L$  when given the sample values  $x_A = \{x_{\alpha_1}, \dots, x_{\alpha_{|A|}}\}$  is  $\sum_{i=1}^{|A|} w_i x_{\alpha_i}$ .*

Intuitively, semilinear algorithms work by mapping the sample/target sets to linear coefficients of the sample values, effectively specifying separate linear estimators for each possible sample/target pair  $(A, B)$  supported on  $P$ . A good semilinear algorithm will produce a set of estimators such that there is no assignment to values  $x$  that incurs high error on many of the possible sample/target sets. Such an algorithm will have low worst-case expected error, which is our goal.

Although the class of semilinear algorithms is a restriction, it is a natural starting point for studying this general framework. Additionally, nearly all estimation algorithms that we are aware of fall into this class. The sample mean is trivially semilinear; the more surprising constant-factor optimal scheme in the selective prediction setting of Example 2 produces a semilinear algorithm where the

weights depend on the target indices, not just on the sample indices (recall that the returned estimate is the average of the  $w$  highest-indexed sample values, where  $w$  is the size of the target set). In the setting of “importance sampling” where each element  $i$  is included in the sample independently, with probability  $p_i$ , standard estimators are also semilinear in that they return a weighted mean of the sample values where the weight of  $x_i$  is typically a function of  $p_i$ . Our notion of semilinear in this setting is even more general, in that the weights given to a sample value  $x_i$  can depend arbitrarily on both  $p_i$ , as well as the other probabilities  $\{p_{j \neq i}\}$  and the specific set of sampled indices.

Our main result is that there exists an efficient algorithm which, given sample-access to the joint sample/target distribution  $P$ , returns a semilinear estimation scheme whose worst-case expected error is within a constant factor of the optimal such scheme for  $P$ .

**Theorem 1.** *Let  $P$  denote a distribution over pairs of subsets  $(A, B)$  of  $\{1, \dots, n\}$ , and let  $\epsilon > 0$  be a fixed error parameter. There is an algorithm  $L$  which, given sample-access to  $P$  and given sets  $A = (\alpha_1, \dots, \alpha_{|A|})$  and  $B = (\beta_1, \dots, \beta_{|B|})$ , takes  $\text{poly}(n, 1/\epsilon)$  samples from  $P$ , runs in time  $\text{poly}(n, 1/\epsilon)$ , and returns a list of  $|A|$  weights,  $w_1^{L(A,B)}, \dots, w_{|A|}^{L(A,B)}$ , with the following guarantee:*

*For any values  $x = \{x_1, \dots, x_n\}$  with  $|x_i| \leq 1$  and with high probability over  $L$ 's samples from  $P$ , the expected squared difference between the estimate  $\sum_{i=1}^{|A|} w_i^{L(A,B)} x_{\alpha_i}$  and the mean of the target set  $\frac{1}{|B|} \sum_{i=1}^{|B|} x_{\beta_i}$  is within an additive  $\epsilon$  and multiplicative  $\pi/2$  factor of the worst-case expected error of the optimal semilinear algorithm. Formally,*

$$\begin{aligned} & \mathbb{E}_{(A,B) \sim P} \left[ \left( \sum_{i=1}^{|A|} w_i^{L(A,B)} x_{\alpha_i} - \frac{1}{|B|} \sum_{i=1}^{|B|} x_{\beta_i} \right)^2 \right] \\ & \leq \epsilon + \frac{\pi}{2} \left( \inf_{L': (A,B) \rightarrow \{w_i^{L'(A,B)}\}} \sup_{(x'_1, \dots, x'_n): |x'_i| \leq 1} \mathbb{E}_{(A,B) \sim P} \left[ \left( \sum_{i=1}^{|A|} w_i^{L'(A,B)} x'_{\alpha_i} - \frac{1}{|B|} \sum_{i=1}^{|B|} x'_{\beta_i} \right)^2 \right] \right). \end{aligned}$$

As a component of our proof of this theorem, we give an efficient algorithm which approximates the worst-case expected error of any semilinear algorithm to within this multiplicative  $\pi/2$  factor (see Proposition 1 for a formal statement). This approximation factor of  $\frac{\pi}{2}$  is optimal in the sense that there exist a semilinear estimator and distribution  $P$  for which estimating the worst-case expected error to within a  $\frac{\pi}{2}$  factor is NP-hard. We discuss this in Section 2.1.

While we focus on semilinear schemes, these are not necessarily optimal (see Appendix D for details):

**Fact 1.** *There exists a distribution  $P$  for which the optimal semilinear mean estimation scheme achieves a worst-case expected squared error that is larger than the worst-case expected squared error of the optimal (unconstrained) scheme.*

As we discuss in Section 1.3, one open question is to understand the severity of this gap between semilinear versus arbitrary estimation algorithms. We conjecture this gap is bounded by a small constant; the most extreme gap that we know of, found via an automated search, is a factor of 1.004.

**Linear Regression.** We also consider the following natural extension of our results to the setting of  $d$ -dimensional linear regression. As above, there is a joint distribution  $P$  over sample/target subsets  $A, B \subset \{1, \dots, n\}$ . Each index  $i$  has an associated labeled vector,  $(x_i, y_i)$  with  $x_i \in \mathbb{R}^d$  and  $y_i \in \mathbb{R}$ . For a sample/target pair  $A, B$ , we observe the data points  $(x_i, y_i)$  for  $i \in A$ , and the task is to return a linear regression model  $\hat{\beta}$  that has small error when applied to the (unobserved) datapoints indexed by the target set  $B$ . Specifically, letting  $\beta_B$  denote the least-squares regression model for the target set  $\{(x_i, y_i)\}_{i \in B}$ , the goal is to return  $\hat{\beta} \approx \beta_B$ .

The following theorem leverages Theorem 1; the algorithm and proof are given in Appendix E.

**Theorem 2.** *Consider the regression setting described above corresponding to a sample/target distribution  $P$ , with the additional guarantee that each coordinate of the features  $x_i$  and labels  $y_i$  are scaled so as to have magnitude at most 1. Let  $\alpha_P$  be the mean squared error guaranteed by Theorem 1 for the (scalar) mean estimation setting. There is a polynomial-time algorithm which, given  $A, B$ , the datapoints  $\{(x_i, y_i)\}_{i \in A}$  and sample-access to  $P$ , with probability  $1 - \delta$  returns*

regression coefficients  $\hat{\beta}$  such that  $\|\hat{\beta} - \beta_B\| \leq 3\sqrt{\frac{\alpha_P d^3/\delta}{\sigma_d^2}}$ , for any  $\delta > 0$ ; here  $\beta_B$  is the true vector of least-squares coefficients for  $\{(x_i, y_i)\}_{i \in B}$ , and  $\sigma_d$  denotes the smallest singular value of the covariance  $\frac{1}{|B|} \sum_{i \in B} x_i x_i^\top$ . In the case that the features  $\{x_i\}_{i \in B}$  are known for the target set (and only the  $y_i$ 's are unknown) then we have the improved bound that, except with  $\delta$  probability,  $\|\hat{\beta} - \beta_B\| \leq \frac{\sqrt{\alpha_P d/\delta}}{\sigma_d}$ .

## 1.2 Related Work

There has been significant recent effort developing algorithms for estimation and learning with strong performance guarantees beyond the idealized setting of data drawn i.i.d. from a fixed distribution. This includes the recent body of work on *robust* learning and statistics. Building off a long line of work from the statistics community (see e.g. [15, 24]), the models considered in these works assume that datapoints are drawn independently from some distribution of interest, and then an  $\alpha$  fraction of datapoints are corrupted arbitrarily/adversarially. (Some of these works also consider the slightly weaker *contamination* model where the  $\alpha$  fraction of arbitrary data is specified before the  $1 - \alpha$  fraction of i.i.d. data is drawn.) Recent work has developed computationally efficient algorithms for basic estimation and learning tasks in these settings, beginning with estimating the mean and covariance of a high-dimensional Gaussian [8, 19], and subsequently considering more general optimization problems over data, including linear regression [4, 23, 18, 16]. While this line of work relaxes the typical assumption that *all* datapoints are drawn i.i.d. from a distribution, these works still rely on the assumption that a significant fraction of the data is drawn from a well-behaved distribution. From a technical perspective, these works typically proceed by analyzing the structure of the  $1 - \alpha$  fraction of i.i.d. datapoints, and then showing that the adversarial datapoints cannot completely obscure this structure. In this sense, the distributional assumptions on the  $1 - \alpha$  fraction of “good” data are critically leveraged.

There is also a line of recent work developing algorithms that work on *truncated* data [6, 7] which captures one commonly arising class of dataset that deviates from the i.i.d. setting. Here, the assumption is that data is drawn independently from a “nice” distribution—a high-dimensional Gaussian in the case of [6]—but then the dataset is truncated, revealing only the portion that lies within some specified set. The challenge is that this conditioning often significantly skews the statistics of the data. Work on learning from truncated samples differs significantly from the framework considered in our work, in that the positive results in [6, 7] leverage assumed structure of the underlying data: the Gaussian assumption in [6], and for [7] the assumptions of an underlying noisy linear model and that the truncation procedure is only a function of the label of each datapoint.

Beyond the dependencies that truncation introduces, recent work also considers regression in a setting with more complex dependencies, that models the type of dependence that may arise when datapoints correspond to nodes within a network [5]. In that work, the authors revisit the standard noisy linear regression model with labels  $y_i = \theta^T x_i + \epsilon_i$ , and the standard logistic regression model where  $Pr[y_i = 1] = 1/(1 + \exp(\theta^T x_i))$ . Instead of assuming that the  $\epsilon_i$  and logistic outcomes are drawn independently, they consider the case where these are generated in a correlated fashion, corresponding to a known/fixed covariance matrix with an unknown strength parameter. Despite these dependencies, the authors provide an efficient algorithm for learning the model,  $\theta$ , in this setting, that still achieves the error guarantees of the independent settings, provided some mild assumptions are satisfied.

Finally, it is worth clarifying the distinction between our framework and the *on-line* learning framework. As with our framework, much of the work in on-line learning makes no assumptions about the underlying data, and often assumes that the underlying data is adaptively responding to our predictions. Beyond this, however, the frameworks are quite different: our framework considers the task of making a single prediction, as opposed to a sequence of predictions. Additionally, we are measuring the performance of algorithms against all algorithms in their class, instead of in comparison to some set of fixed benchmarks.

## 1.3 Discussion and Open Directions

We introduce a framework for understanding statistical estimation where we make no assumptions on the data values themselves, but model the process  $P$  by which sample and target datasets are collected. Within this framework, an estimator is evaluated based on its worst-case expected error,

where the worst-case is with respect to the data values, and the expectation is with respect to the selection of the sample and target sets according to  $P$ . We present algorithms for approximating the worst-case expected error of an estimator with respect to  $P$ , and for computing an estimator that approximately minimizes this error, within a broad class of estimators. In addition to the strong theoretical guarantees, this algorithm yields estimators that seems to perform extremely well on several natural synthetic settings where samples are drawn from nontrivial sampling distributions.

There are a number of natural open directions prompted by this work. Can the algorithms be adapted to have a runtime independent of  $n$ ? The framework and definition of worst-case expected error certainly applies to the case where the underlying domain is infinite (instead of  $\{1, \dots, n\}$ )—can efficient algorithms be developed for that setting? If data values are constrained to an  $\ell_1$  or  $\ell_2$  ball (as opposed to the  $\ell_\infty$  ball corresponding to our assumption that each value is bounded), are simpler algorithms possible? What is the gap between the worst-case expected error of the best semilinear mean estimation algorithm and the best unconstrained scheme? It also seems worth considering specific classes of distributions from the perspective offered by our framework. For example, for  $P$  corresponding to snowball sampling over a social network, what network properties imply subconstant estimation error? Are there variants of snowball sampling that yield significantly better or worse values of the expected estimation error for worst-case data?

Finally, it seems worthwhile extracting high-level interpretable properties of  $P$  that imply the existence of estimators with subconstant (or inverse polynomial) error, or properties that imply that the worst-case expected error will be constant for any estimator. For the many cases where we have control over how data is collected, such properties could serve to guide the design of these data collection pipelines.

## 2 Algorithms and Connection to the Grothendieck Problem

Given elements  $\{1, \dots, n\}$  from which both the sample and target sets are drawn, the main component of our model is a joint distribution  $P$  on the sample and target sets  $(A, B)$ . Such a joint distribution can be approximated to arbitrary accuracy as an unweighted distribution over a list of pairs  $(A_i, B_i)$ , and for ease of notation, we adopt this representation here.

**Definition 2.** *A joint sample/target distribution over a universe of  $n$  elements is specified by a list  $P$  of pairs  $(A_i, B_i)$  of some length  $m$ , where for each  $i \in \{1, \dots, m\}$  the sets  $A_i, B_i$  are subsets of  $\{1, \dots, n\}$ . To sample from this distribution, choose a uniformly random  $i \sim \{1, \dots, m\}$  and let  $A_i$  be the sample indices, and  $B_i$  be the target indices. For values  $\{x_1, \dots, x_n\}$ , the sample values will be  $x_{A_i}$  and the target values will be  $x_{B_i}$ .*

Given a sample set  $A$  and target set  $B$  from such a distribution  $P$ , for values  $x_1, \dots, x_n$  the algorithmic challenge is to predict a desired attribute of the target values  $x_B$ , using knowledge of the sample values  $x_A$  along with the indices  $A$  and  $B$ . We first focus on the case where the values are real numbers and the goal is to compute the arithmetic mean of the target values. Our objective is to minimize the root-mean-squared error of the estimate of the mean, relative to the scale of the data,  $\max_i |x_i|$ . Since semilinear estimators scale linearly with the data, this is equivalent to normalizing the data by dividing through by  $\max_i |x_i|$ , and then minimizing the mean squared error subject to the data bound  $|x_i| \leq 1$ , which is the approach we adopt throughout.

**Definition 3.** *For an estimation algorithm  $f(x_A, A, B)$  taking as inputs the sample values along with the indices of the sample and target sets, we define the worst-case expected performance for distribution  $P$ , to be the maximum over values  $x_1, \dots, x_n$  of the mean squared error of its estimate:*

$$\max_{x_1, \dots, x_n \in [-1, 1]} \frac{1}{m} \sum_{i=1}^m (f(x_{A_i}, A_i, B_i) - \text{mean}(x_{B_i}))^2.$$

For *semilinear* estimators the following notation will simplify the analysis.

**Definition 4.** *Given a sample/target distribution  $P = (A_1, B_1), \dots, (A_m, B_m)$ , a semilinear estimation algorithm consists of a vector  $a_i \in \mathbb{R}^n$  for each  $i \in \{1, \dots, m\}$ , where the support of  $a_i$  is a subset of  $A_i$ . Thus the estimate  $f(x_{A_i}, A_i, B_i)$  is simply evaluated as the vector-vector product  $a_i^T x$ .*

*Correspondingly, we reexpress  $\text{mean}(x_{B_i}) = b_i^T x$  by defining for each  $B_i$  a corresponding vector  $b_i \in \mathbb{R}^n$ , where  $b_i(j) = \frac{1}{|B_i|}$  for  $j \in B_i$  and 0 otherwise. Thus the performance of a semilinear*

estimation algorithm ( $a_i$ ) equals

$$\max_{x_1, \dots, x_n \in [-1, 1]} \frac{1}{m} \sum_{i=1}^m ((a_i - b_i)^T x)^2 = \max_{x_1, \dots, x_n \in \{-1, 1\}} x^T \left( \frac{1}{m} \sum_{i=1}^m (a_i - b_i)^T (a_i - b_i) \right) x. \quad (1)$$

In the second expression above, we (equivalently) restrict the range of each  $x_j$  to the endpoints  $\{-1, 1\}$  since the expression being maximized is a positive semidefinite quadratic form of  $x$ , and thus each  $x_j$  may be moved to one of the endpoints of its range without decreasing the objective function.

**Definition 5.** Given a sample/target distribution  $P = (A_1, B_1), \dots, (A_m, B_m)$ , the performance of the best semilinear estimator is

$$\frac{1}{m} \min_{a_i: \{j: a_i(j) \neq 0\} \subseteq A_i} \max_{x_1, \dots, x_n \in \{-1, 1\}} \sum_{i=1}^m ((a_i - b_i)^T x)^2.$$

## 2.1 Worst-Case Performance of a Fixed Semilinear Estimator

Here we consider the challenge of optimizing Equation 1: given a fixed semilinear estimator, how good is it? As noted above, the matrix in the parentheses in the second expression of Equation 1 is positive semidefinite. In fact, for appropriate coefficients  $a_i$ , we can make the matrix  $\frac{1}{m} \sum_{i=1}^m (a_i - b_i)^T (a_i - b_i)$  be an arbitrary positive semidefinite matrix (though possibly at the cost of an “unnatural” estimation algorithm). Thus the problem of computing or estimating the performance of a given estimation algorithm is identical to what is known as the positive semidefinite Grothendieck problem. Since the Grothendieck problem includes MAX-CUT, which is NP-hard, evaluating the performance of a fixed estimator is also NP-hard. Further, as was recently shown by Briët, Regev, and Saket, it is NP-hard even to approximate the semidefinite Grothendieck problem to within a factor of  $\frac{\pi}{2}$  [17, 3]. Thus, even for a fixed semilinear estimator, we cannot hope to approximate its performance—given by Equation 1—to better than a factor of  $\frac{\pi}{2}$  in polynomial time.

However, analogously to the Goemans-Williamson semidefinite relaxation of MAX-CUT, we consider the semidefinite relaxation of the semidefinite Grothendieck problem, replacing each scalar variable  $x_j$  with a vector  $v_j$  in the  $n$ -dimensional unit ball. The proof of the following proposition is given in a self-contained fashion in Appendix A, and is based on the analysis of the randomized rounding scheme from Nesterov [20]. In this appendix we also provide some additional explanation and background on the Grothendieck problem and relaxation.

**Proposition 1.** Given a sample/target distribution  $P = (A_1, B_1), \dots, (A_m, B_m)$ , the problem of evaluating the performance  $p$  of a semilinear estimator, specified by vectors  $a_1, \dots, a_m \in \mathbb{R}^n$ , is NP-hard to estimate to within a multiplicative factor of  $\frac{\pi}{2}$ . However, letting  $M = \frac{1}{m} \sum_{i=1}^m (a_i - b_i)(a_i - b_i)^T$ , the optimum of the convex (semidefinite) program

$$\max_{V \text{ psd}, V_{(j,j)} \leq 1} \sum_{j,k=1}^n M_{(j,k)} V_{(j,k)} \quad (2)$$

is in the interval  $[p, \frac{\pi}{2}p]$ , and can be found in polynomial time by semidefinite programming.

## 2.2 Computing a Near-Optimal Semilinear Estimator

While Section 2.1 analyzed the problem of evaluating the performance of a fixed semilinear estimator, here instead we aim to find a near-optimal semilinear estimator. This is a challenging setting for optimization, as even evaluating the objective function, to within a factor of  $\frac{\pi}{2}$ , is NP-hard (as discussed in Section 2.1). However, as we will see, the convex (semidefinite) relaxation derived in Section 2.1 not only lets us approximate the performance of a fixed estimator to a  $\frac{\pi}{2}$  factor, but also provides the crucial structure enabling us to find a semilinear estimator whose performance is within a  $\frac{\pi}{2}$  factor of the best possible semilinear estimator.

**Theorem 3.** Algorithm 1, given a description of the joint distribution of sample and target sets  $(A_1, b_1), \dots, (A_m, b_m)$ , runs in polynomial time, and returns coefficients for a semilinear estimator whose expected squared error is within a  $\frac{\pi}{2}$  factor of that of the best semilinear estimator. The value of the objective function achieved by  $\hat{V}$  is  $m$  times the Proposition 1 SDP bound on the mean squared error of the best semilinear estimator.

---

**Algorithm 1** SDP Algorithm yielding  $\frac{\pi}{2}$ -approximation to the best semilinear estimator

---

**Input:** A joint distribution  $P$  of sample and target sets, expressed as a list of pairs  $(A_1, b_1), \dots, (A_m, b_m)$ , where each  $A_i \subset \{1, \dots, n\}$  is the indices of the sample set in the  $i^{\text{th}}$  pair, and each  $b_i$  is a vector with uniform values over the target set in the  $i^{\text{th}}$  pair, as in Definition 4.

For an  $n \times n$  matrix  $V$  and a set  $A_i \in \{1, \dots, n\}$ , let  $V_{A_i}$  denote  $V$  restricted to the rows in  $A_i$ , and let  $V_{A_i, A_i}$  denote  $V$  restricted to *both* rows and columns in  $A_i$ .

1. Compute the concave maximization

$$\widehat{V} = \arg \max_{V \text{ psd}, V_{j,j} \leq 1} \sum_{i=1}^m b_i^T (V - V_{(A_i)}^T V_{(A_i, A_i)}^{-1} V_{(A_i)}) b_i \quad (3)$$

2. For each  $i \in \{1, \dots, n\}$ , let  $a_i$  when restricted to the coordinates  $A_i$  equal  $\widehat{V}_{(A_i, A_i)}^{-1} \widehat{V}_{(A_i)} b_i$ , and 0 on the remaining coordinates. **Output**  $a_1, \dots, a_m$ .
- 

The function inside the sum in Step 1 can be reexpressed as a standard semidefinite program, using the Schur complement to reexpress the matrix inverse—which is done automatically, for example, in the CVXPY package, as used in our code for the experiments in Section 3. The inverse in Step 2 to compute the linear coefficients can be interpreted as a pseudoinverse  $\widehat{V}_{(A_i, A_i)}^+$  in cases where it would otherwise be singular.

The proof of Theorem 3 is given in Appendix B. While Algorithm 1 takes as input the entire description of the joint sample/target distribution, such a description might be (1) unavailable in practice and/or (2) have support size  $m$  that is exponentially large. In Appendix C we modify this algorithm to use only a polynomial number of *samples* from distribution  $P$ , addressing both issues mentioned above, establishing Theorem 1.

### 3 Empirical Evaluation

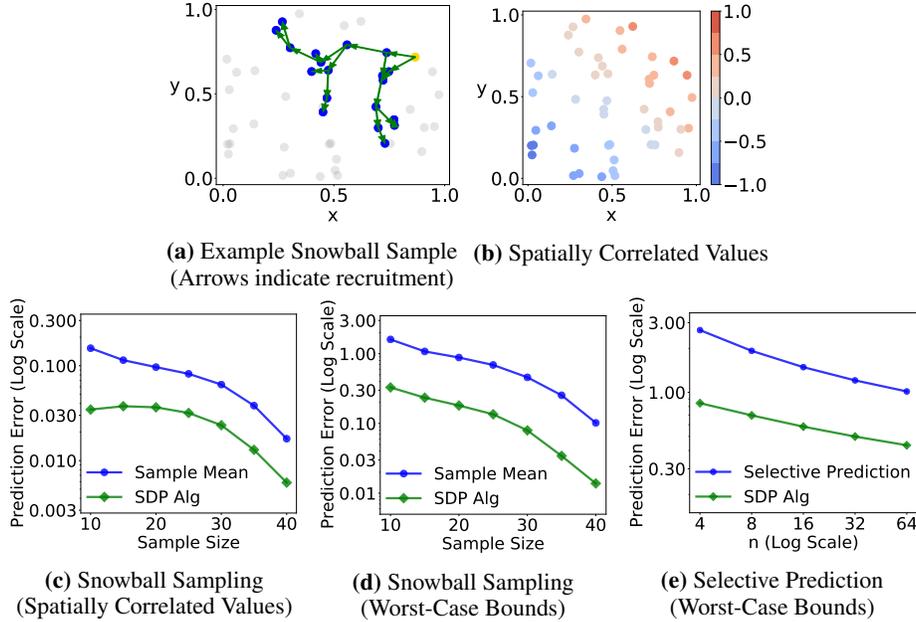
We illustrate the empirical performance of Algorithm 1 for estimating the target mean in three natural settings in which membership in the sample may be perniciously correlated with the underlying data values. The SDP formulation for estimating the worst-case expected squared error of a fixed semilinear estimator in Proposition 1 allows us to compare our algorithm with natural baseline estimators. These experiments are based on our implementation of Algorithm 1 using the Python CVXPY package [9, 1] with the MOSEK solver [2]—our code is available at <https://github.com/justc2/worst-case-randomly-collected>.

**Importance Sampling:** We consider a set of  $n = 50$  elements where the  $i$ th element is included in the sample set independently with probability  $p_i$ , with  $p_1, \dots, p_{25} = 0.1$  and  $p_{26}, \dots, p_{50} = 0.5$ . The target set is the entire population, i.e. the goal is to estimate the population mean. Table 1 compares the performance of the estimator given by our framework with two standard baselines: *inverse proportional reweighting* and *subgroup estimation*. For inverse proportional reweighting, the estimate is given by the weighted mean of the values in the sample set where the weight for value  $x_i$  is  $1/p_i$ . For subgroup estimation, the estimator separately computes the sample mean for  $i \leq 25$  and  $i > 25$  and then returns the average of these two values. We empirically evaluate the above estimators in terms of expected squared error on the following data values, which are designed to illustrate the strengths and weaknesses of the above estimators. For *constant* values, all values are 1; for *intergroup variance*,  $x_1, \dots, x_{25} = 1$  and  $x_{26}, \dots, x_{50} = -1$ ; for *intragroup variance*,  $x_{\text{even}} = 1$  and  $x_{\text{odd}} = -1$ ; and for *worst-case*, the approximate worst case error is given by solving the SDP relaxation described in Section 2.1 (note that in this setting the approximation error is negligible).

Even in this simple setting with independent sampling, our algorithm (“SDP Alg”) gives roughly a two-fold improvement over the baselines in terms of worst-case expected error. While both inverse proportional reweighting and subgroup estimation are unbiased, they have high variance in some of the settings. As inverse proportional reweighting is non-adaptive to the sample size (it always assigns the same weights to the elements), the size of the sample set has a large impact on bias of the estimate. Subgroup estimation is adaptive, but if we only get a few samples from one of the groups, it

Data Values	Inv. Prop. Reweighting	Subgroup Estimation	SDP Alg
Constant ( $x_i = 1$ )	0.100	0.018	0.051
Intergroup Variance	0.100	0.018	0.053
Intragroup Variance	0.100	0.121	0.052
Worst-Case Bound (via SDP)	0.101	0.122	0.053

**Table 1:** Importance Sampling Experiment: Comparison of expected squared errors of our approach (SDP Alg) and two standard unbiased estimators, across three different assignments to underlying data, together with worst-case bounds given by our SDP. See text for details of the setting.



assigns high weights to those samples, which contribute a high variance. SDP Alg is adaptive to the sampling process, keeping the error small in all cases, by effectively regularizing its estimate to avoid over-reliance on any small number of elements.

**Snowball/Chain Sampling:** In this experiment, an underlying set of  $n = 50$  points is drawn uniformly from the 2D unit square. A sample is generated by first including a randomly selected point; then, iteratively, each point in the sample “recruits” two of its five closest neighbors, until the desired sample size is reached. The upper left pane of the figure depicts this process. The target set is the entire population of  $n$  individuals. We compare the average squared error of our mean estimation algorithm (labeled SDP Alg) versus that of the naive estimator that returns the sample mean. We consider two settings: 1) the true values are spatially-correlated and are given by the sum of the  $x$  and  $y$  coordinates of the point (bottom left pane), and 2) the worst-case values for this sampling distribution, approximated by the SDP relaxation from Section 2.1 (bottom middle pane). In both cases, our algorithm outperforms the baseline estimator: by 2–4 $\times$  for spatially-correlated values (even though our algorithm was not optimizing for this case!) and by 4–7 $\times$  for worst-case values.

**Selective Prediction:** The selective prediction setting, described in Example 2 and considered in [10, 21], corresponds to chronological prediction problems in which samples from the past are used to make estimates about data in the future. These previous works show that the estimator which, to predict the mean of a target set of size  $w$  outputs the mean of the most “recent”  $w$  points in the sample, achieves worst-case expected squared error  $O(\frac{1}{\log n})$ , where  $n$  is the total number of elements, and that this is asymptotically optimal. The bottom right pane of the figure compares that estimator with our algorithm, illustrating that both approaches scale with  $O(\frac{1}{\log n})$  but that our algorithm yields a 2–3 $\times$  reduction in error.

## Broader Impact

The question of how to extract accurate statistics based on nonuniform/biased samples is of utmost societal importance. And this basic question is still far from solved—one need only look to the consistent errors across political polls, or more recent discussion on estimating the rate of COVID exposure based on different strategies for recruiting participants and then “correcting” for these biased samples. The vast majority of work on accurate estimation is based on strong distributional assumptions on the data values. The risk is that when these assumptions do not hold, the estimates and their confidence bounds, are meaningless. In this work, we introduce a very general framework that allows one to ask (and answer) the question of whether a given data collection procedure can admit an estimation algorithm which will be accurate, even for worst-case data values. We hope that this framework, which we refer to as *worst case analysis for randomly collected data*, will offer better estimators in some settings, and offer new perspectives on collecting and inferring information from data samples.

## Acknowledgments and Disclosure of Funding

Justin Chen is partially supported by the NSF Graduate Research Fellowship under Grant No. 1745302, the NSF CCF-2006806 award, and the Simons Investigator Award. Gregory Valiant is partially supported by NSF awards AF-1813049, CCF-1704417, and 1804222, ONR Young Investigator Award N00014-18-1-2295, and DOE Award DE-SC0019205. Paul Valiant is partially supported by NSF awards IIS-1562657, DMS-1926686, and indirectly supported by NSF award CCF-1900460.

## References

- [1] Akshay Agrawal, Robin Verschueren, Steven Diamond, and Stephen Boyd. A rewriting system for convex optimization problems. *Journal of Control and Decision*, 5(1):42–60, 2018.
- [2] MOSEK ApS. *MOSEK Optimizer API for Python 9.2.10*, 2019.
- [3] Jop Briët, Oded Regev, and Rishi Saket. Tight hardness of the non-commutative Grothendieck problem. *Theory of Computing*, 13(15):1–24, 2017.
- [4] Moses Charikar, Jacob Steinhardt, and Gregory Valiant. Learning from untrusted data. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 47–60. ACM, 2017.
- [5] Constantinos Daskalakis, Nishanth Dikkala, and Ioannis Panageas. Regression from dependent observations. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 881–889. ACM, 2019.
- [6] Constantinos Daskalakis, Themis Gouleakis, Christos Tzamos, and Manolis Zampetakis. Efficient statistics, in high dimensions, from truncated samples. In *59th Annual IEEE Symposium on Foundations of Computer Science*, 2018.
- [7] Constantinos Daskalakis, Themis Gouleakis, Christos Tzamos, and Manolis Zampetakis. Computationally and statistically efficient truncated regression. *Proceedings of Machine Learning Research vol XX*, 1:31, 2019.
- [8] Ilias Diakonikolas, Gautam Kamath, Daniel Kane, Jerry Li, Ankur Moitra, and Alistair Stewart. Robust estimators in high-dimensions without the computational intractability. *SIAM Journal on Computing*, 48(2):742–864, 2019.
- [9] Steven Diamond and Stephen Boyd. CVXPY: A Python-embedded modeling language for convex optimization. *Journal of Machine Learning Research*, 17(83):1–5, 2016.
- [10] Andrew Drucker. High-confidence predictions under adversarial uncertainty. *ACM Transactions on Computation Theory (TOCT)*, 5(3):12, 2013.

- [11] Michel X. Goemans and David P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *J. ACM*, 42(6):1115–1145, November 1995.
- [12] Leo A Goodman. Snowball sampling. *The annals of mathematical statistics*, pages 148–170, 1961.
- [13] Johan Håstad. Some optimal inapproximability results. *J. ACM*, 48(4):798–859, July 2001.
- [14] Douglas D Heckathorn. Respondent-driven sampling ii: deriving valid population estimates from chain-referral samples of hidden populations. *Social problems*, 49(1):11–34, 2002.
- [15] Peter J Huber et al. The behavior of maximum likelihood estimates under nonstandard conditions. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 221–233. University of California Press, 1967.
- [16] Sushrut Karmalkar, Pravesh Kothari, and Adam Klivans. List-decodable linear regression. *arXiv preprint arXiv:1905.05679*, 2019.
- [17] Subhash Khot and Assaf Naor. Approximate kernel clustering. *Mathematika*, 55(1-2):129–165, 2009.
- [18] Adam Klivans, Pravesh K Kothari, and Raghu Meka. Efficient algorithms for outlier-robust regression. In *Conference On Learning Theory*, pages 1420–1430, 2018.
- [19] Kevin A Lai, Anup B Rao, and Santosh Vempala. Agnostic estimation of mean and covariance. In *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 665–674. IEEE, 2016.
- [20] Yuri Nesterov. Semidefinite relaxation and nonconvex quadratic optimization. *Optimization Methods and Software*, 9(1-3):141–160, 1998.
- [21] Mingda Qiao and Gregory Valiant. A theory of selective prediction. In *Conference on Learning Theory (COLT)*, 2019.
- [22] Maurice Sion. On general minimax theorems. *Pacific Journal of mathematics*, 8(1):171–176, 1958.
- [23] Jacob Steinhardt, Moses Charikar, and Gregory Valiant. Resilience: A criterion for learning in the presence of arbitrary outliers. In *9th Innovations in Theoretical Computer Science Conference (ITCS 2018)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018.
- [24] John W Tukey. Mathematics and the picturing of data. In *Proceedings of the International Congress of Mathematicians, Vancouver, 1975*, volume 2, pages 523–531, 1975.

## A Proof of Proposition 1

In this section we give a self-contained proof of Proposition 1, restated here for convenience:

**Proposition 1** *Given a sample/target distribution  $P = (A_1, B_1), \dots, (A_m, B_m)$ , the problem of evaluating the performance  $p$  of a semilinear estimation algorithm specified by vectors  $a_1, \dots, a_m \in \mathbb{R}^n$  is NP-hard to estimate to within a multiplicative factor of  $\frac{\pi}{2}$ . However, letting  $M = \frac{1}{m} \sum_{i=1}^m (a_i - b_i)^T (a_i - b_i)$ , the optimum of the convex (semidefinite) program*

$$\max_{V \text{ psd}, V_{(j,j)} \leq 1} \sum_{j,k=1}^n M_{(j,k)} V_{(j,k)}$$

*is in the interval  $[p, \frac{\pi}{2}p]$ , and can be found in polynomial time by standard semidefinite programming algorithms.*

The proof of this proposition leverages the connection to the positive semidefinite Grothendieck problem:

**Definition 6.** *The positive semidefinite Grothendieck problem, given an  $n \times n$  positive semidefinite matrix  $M$  is to evaluate:*

$$\max_{x_1, \dots, x_n \in \{-1, 1\}} x^T M x \quad (4)$$

*(Note that this problem is sometimes phrased as the optimization over a pair of vectors  $x, y$ , of the expression  $x^T M y$ , though for positive semidefinite  $M$ , an optimum will always be attained when  $x = y$ .)*

The positive semidefinite Grothendieck problem includes MAX-CUT as a special case, since, for an undirected graph  $G$ , its Laplacian  $L$  is positive semidefinite, and for any vector  $x \in \{-1, 1\}^n$  that labels its vertices, the value of  $x^T L x$  will equal the total degree of the graph plus the size of the cut induced by the labels of  $x$ . Thus, since MAX-CUT is NP-hard, evaluating the performance of a fixed estimator is also NP-hard. Further, Håstad showed that it is NP-hard to even approximate MAX-CUT to within a multiplicative factor of  $\frac{17}{16}$  [13]. For the more general case of the semidefinite Grothendieck problem considered here, Khot and Naor showed the unique-games hardness of approximating the optimum to within a factor of  $\frac{\pi}{2}$ ; this result was recently strengthened by Briët, Regev, and Saket to show it is in fact NP-hard to get an approximation ratio better than  $\frac{\pi}{2}$  [17, 3]. Thus, even for a fixed semilinear estimation algorithm, we cannot hope to approximate its performance—given by Equation 1—to within a factor of  $\frac{\pi}{2}$ .

Analogously to the Goemans-Williamson semidefinite relaxation of MAX-CUT, we consider the semidefinite relaxation of the semidefinite Grothendieck problem, replacing each scalar variable  $x_j$  with a vector  $v_j$  in the  $n$ -dimensional unit ball.

**Definition 7.** *Given an  $n \times n$  positive semidefinite matrix  $M$ , the semidefinite relaxation of the positive semidefinite Grothendieck problem is to evaluate:*

$$\max_{v_j \in \mathbb{R}^n: \|v_j\| \leq 1} \sum_{j,k=1}^n M_{(j,k)} (v_j^T v_k) \quad (5)$$

*or, equivalently, letting “psd” denote the property of a matrix being positive semidefinite,*

$$\max_{V \text{ psd}, V_{(j,j)} \leq 1} \sum_{j,k=1}^n M_{(j,k)} V_{(j,k)}$$

Crucially, the set of positive semidefinite matrices is convex, so thus the optimization problem of Definition 7 (in its second form) maximizes a linear function over a convex set, and thus can be computed in polynomial time.

Goemans and Williamson famously showed, via a randomized rounding scheme, that the gap between MAX-CUT and the result of the induced positive semidefinite relaxation is bounded by a factor of 1.14 [11]. For the more general setting here, of arbitrary positive semidefinite matrices instead of graph Laplacians, Nesterov showed a bound of  $\frac{\pi}{2}$  [20]. We include a self-contained derivation here, for the sake of completeness.

Since scaling a single vector  $v_j$  affects Equation 5 in a convex quadratic manner, there will always be an optimum of Equation 5 where  $\|v_j\| = 1$  for all  $j$ . We assume this, for simplicity, when describing the randomized rounding procedure below.

**Definition 8.** Given  $n$  unit vectors  $v_j \in \mathbb{R}^n$ , for  $j \in \{1, \dots, n\}$ , the Goemans-Williamson randomized rounding procedure chooses a random direction  $r$ , and for each vector  $v_j$  returns a scalar  $x_j = \text{sign}(r^T v_j)$ .

**Proposition 2.** Given an  $n \times n$  positive semidefinite matrix  $M$ , and  $n$  unit vectors  $v_1, \dots, v_n \in \mathbb{R}^n$ , the value of the relaxed Grothendieck problem,  $\sum_{j,k=1}^n M_{(j,k)}(v_j^T v_k)$  is at most  $\frac{\pi}{2}$  times the expected value of the original Grothendieck problem evaluated on scalars  $x_1, \dots, x_n \in [-1, 1]$  obtained from  $v_1, \dots, v_n$  by the Goemans-Williamson randomized rounding procedure,  $\mathbb{E}[\sum_{j,k=1}^n M_{(j,k)} x_j x_k]$ .

Thus for any objective value that can be achieved in the relaxed problem, with vectors  $v_1, \dots, v_n$ , the original problem can achieve an objective value at least a  $\frac{2}{\pi}$  fraction of it, since it does so in expectation over scalars  $x_1, \dots, x_n$  obtained by the randomized rounding procedure.

*Proof of Proposition 2.* As in the analysis of the Goemans-Williamson randomized rounding scheme for MAX-CUT, note that the expected value  $\mathbb{E}[x_j x_k] = \mathbb{E}_r[\text{sign}(r^T v_j) \text{sign}(r^T v_k)]$ , where  $r$  is a randomly chosen direction. Because of the rotational symmetry of the distribution of  $r$ , we may equivalently rotate  $v_j$  and  $v_k$  into the plane, from which we can see that, for  $r$  also projected into the plane,  $\text{sign}(r^T v_j) \text{sign}(r^T v_k)$  equals 1 when  $r$  is within  $\frac{\pi}{2}$  radians of *both*  $v_j, v_k$  or *neither* of them. For a randomly chosen  $r$  in the plane, this happens with probability  $1 - \frac{1}{\pi} \theta_{j,k}$ , where  $\theta_{j,k}$  is the angle between  $v_j, v_k$ , yielding that  $\mathbb{E}[x_j x_k] = 1 - \frac{2}{\pi} \theta_{j,k}$ .

As  $\theta_{j,k}$  may be computed as  $\arccos(v_j^T v_k)$ , we may express the expected objective value after randomized rounding as

$$\mathbb{E}\left[\sum_{j,k=1}^n M_{(j,k)} x_j x_k\right] = \sum_{j,k=1}^n M_{(j,k)} \left(1 - \frac{2}{\pi} \arccos(v_j^T v_k)\right)$$

Recall our overall aim, to show that this value times  $\frac{\pi}{2}$  is greater than or equal to  $\sum_{j,k=1}^n M_{(j,k)}(v_j^T v_k)$ . Subtracting these two quantities means that we need to show that the following quantity is nonpositive:

$$\sum_{j,k=1}^n M_{(j,k)} \left(v_j^T v_k - \frac{\pi}{2} + \arccos(v_j^T v_k)\right) \quad (6)$$

The power series expansion of  $\arccos(y)$  starts  $\arccos(y) = \frac{\pi}{2} - y + \sum_{\ell \geq 3} c_\ell y^\ell$  where all the remaining coefficients  $c_\ell$  are nonpositive, and converges on the entire interval  $y \in [-1, 1]$ . Thus Equation 6 equals

$$\sum_{j,k=1}^n \left( M_{(j,k)} \sum_{\ell \geq 3} c_\ell (v_j^T v_k)^\ell \right) \quad (7)$$

Since the matrix with  $(j, k)$  entry  $v_j^T v_k$  is positive semidefinite for any vectors  $v_1, \dots, v_n$ , and since elementwise raising a positive semidefinite matrix to a positive integer power yields another positive semidefinite matrix, Equation 7 can be reexpressed as  $\sum_{\ell \geq 3} \sum_{j,k=1}^n M_{(j,k)} N_{(j,k)}^{(\ell)}$  for some *negative* semidefinite matrices  $N^{(\ell)}$ , which is thus clearly less than or equal to 0, as desired.  $\square$

Combining the lower bounds and upper bounds of this section immediately yields Proposition 1.

## B Proof of Theorem 3

For ease of exposition, we restate the Theorem 3:

**Theorem.** *Algorithm 1, given a description of the joint distribution of sample and target sets  $(A_1, b_1), \dots, (A_m, b_m)$ , runs in polynomial time, and returns coefficients for a semilinear estimator whose expected squared error is within a  $\frac{\pi}{2}$  factor of that of the best semilinear estimator. The value of the objective function achieved by  $\hat{V}$  is  $m$  times the Proposition 1 SDP bound on the mean squared error of the best semilinear estimator.*

*Proof.* Proposition 1 describes a convex optimization problem to approximate to within a factor of  $\frac{\pi}{2}$  the performance of an estimator specified by vectors  $a_1, \dots, a_m \in \mathbb{R}^n$ . We thus consider optimizing Equation 2 over this choice (omitting the  $\frac{1}{m}$  factor for convenience):

$$\min_{a_i: \{j: a_i(j) \neq 0\} \subseteq A_i} \max_{V \text{ psd}, V_{(j,j)} \leq 1} \sum_{i=1}^m \sum_{j,k=1}^n (a_i - b_i)_{(j)} (a_i - b_i)_{(k)} V_{(j,k)} \quad (8)$$

By Proposition 1, this minimum (if we can efficiently find it), will be within a factor of  $\frac{\pi}{2}$  of the performance of the best semilinear estimator, and the vectors  $a_1, \dots, a_m$  that achieve this minimum will describe an estimator with this performance.

We proceed by invoking von Neumann's minimax theorem.

**Fact 2.** *Given a function  $f(x, y)$  that is convex as a function of its first argument and concave as a function of its second argument, and given convex domains  $X, Y$ , at least one of which is bounded, then*

$$\min_{x \in X} \max_{y \in Y} f(x, y) = \max_{y \in Y} \min_{x \in X} f(x, y)$$

The condition that "at least one of  $X, Y$  is bounded" is a relaxation of the original minimax theorem, shown sufficient by Sion [22].

We observe now that all the conditions of the minimax theorem are satisfied by the expression in Equation 8. As a function of  $a_i$ , the expression being optimized is the quadratic form with coefficients specified by the positive semidefinite matrix  $V$ ; thus the expression is a convex function of  $a_i$ , and since such functions are summed over all  $i \in \{1, \dots, m\}$ , the expression is a convex function of all the vectors  $a_1 \dots, a_m$ . Since the expression is *linear* in  $V$ , it is thus also concave as a function of  $V$ . Finally, the domains of the vectors  $a_1 \dots, a_m$ , along with the matrix  $V$  are both convex, and, since a positive semidefinite matrix must have each entry bounded by the size of the largest diagonal entry, the condition that  $V$  has diagonal entries bounded by 1 induces the same bound on the size of all entries of  $V$ .

Thus we invoke the minimax theorem to conclude that the value of Equation 8 is unchanged if we reverse the order of the min and the max:

$$\max_{V \text{ psd}, V_{(j,j)} \leq 1} \min_{a_i: \{j: a_i(j) \neq 0\} \subseteq A_i} \sum_{i=1}^m \sum_{j,k=1}^n (a_i - b_i)_{(j)} (a_i - b_i)_{(k)} V_{(j,k)} \quad (9)$$

Crucially, now, the inner minimization is simply a sum of positive semidefinite quadratic forms in each of the vectors  $a_1, \dots, a_m$ . Reexpressing the inner sum in vector notation as  $(a_i - b_i)^T V (a_i - b_i)$ , the gradient of this quadratic form with respect to  $a_i$  equals  $2V(a_i - b_i)$ . Thus, subject to the constraint that  $a_i$  can only be nonzero on coordinates in  $A_i$ , if there exists a vector  $a_i$  such that  $V(a_i - b_i) = 0$  on coordinates  $A_i$ , then this  $a_i$  attains the minimum; and otherwise the minimum is  $-\infty$ . The solution for  $a_i$ , restricted to the coordinates  $A_i$  is thus  $V_{(A_i, A_i)}^{-1} V_{(A_i)} b_i$  (or, when  $V_{(A_i, A_i)}$  is singular,  $V_{(A_i, A_i)}^+ V_{(A_i)} b_i$  is the least-squares solution). Plugging this  $a_i$  into the quadratic form yields  $b_i^T (V - V_{(A_i)}^T V_{(A_i, A_i)}^{-1} V_{(A_i)}) b_i$  for the inner minimization of the  $i^{\text{th}}$  term of the objective function. Finally, because of the setup of the minimax theorem, this expression must be a concave function of  $V$ , letting us conclude that Algorithm 1 can in fact conduct the optimization in polynomial time, as desired.

(As a side note, directly proving the above objective function is concave is a bit strange; it is a consequence of the fact that for positive definite  $V$ , and vectors  $x$ , the expression  $x^T M^{-1} x$  is convex as a function of *both* arguments, implying it is convex even when both arguments are affine functions of the optimization variables.)  $\square$

## C Proof of Theorem 1: A Sample-Efficient Algorithm for Near-Optimal Semilinear Estimators

While Algorithm 1 takes as input the entire description  $(A_1, B_1), \dots, (A_m, B_m)$  of the joint sample/target distribution, such a description might be (1) unavailable in practice and/or (2) have support  $m$  that is exponentially large. To address both cases, in this section we design an algorithm that achieves essentially the performance guarantees of Algorithm 1 (as given by Theorem 3), though relying only on *sampling* access to  $P$ . Algorithm 2 will run in time polynomial in  $n$  and *independent* of the (possibly exponential) distribution description length  $m$ .

---

**Algorithm 2** Sampling algorithm to approximate the best semilinear estimator

---

**Input:** Accuracy parameter  $\epsilon > 0$ ;  $t$  random samples from the joint distribution of sample and target sets,  $(A_{s_1}, b_{s_1}), \dots, (A_{s_t}, b_{s_t})$ , where each  $A_i \subset \{1, \dots, n\}$  is the set of sample set indices in the  $i^{\text{th}}$  case and each  $b_i$  is a vector with uniform values over the target set in the  $i^{\text{th}}$  case as in Definition 4; and the actual instance to predict, specified by  $(A, b)$  and the values  $x_A$ .

For an  $n \times n$  matrix  $V$  and a set  $A_i \in \{1, \dots, n\}$ , let  $V_{A_i}$  denote  $V$  restricted to the rows in  $A_i$ , and let  $V_{A_i, A_i}$  denote  $V$  restricted to *both* rows and columns in  $A_i$ .

1. Compute the concave maximization

$$\tilde{V} = \arg \max_{V \succeq \epsilon, V_{j,j} \leq 1} \sum_{i=1}^t b_{s_i}^T (V - V_{(A_{s_i})}^T V_{(A_{s_i}, A_{s_i})}^{-1} V_{(A_{s_i})}) b_{s_i} \quad (10)$$

2. Output the estimate  $x_A \tilde{V}_{(A,A)}^{-1} \tilde{V}_{(A)} b$ .
- 

As compared with Algorithm 1, Algorithm 2 restricts the domain of optimization to matrices  $V$  that have eigenvalues at least  $\epsilon$ , instead of at least 0 (which is a convex restriction). Crucially, instead of summing over all  $m$  possible sample/target set possibilities, the optimization is over a small subset of size  $t$ , obtained by sampling. Finally, the output of this algorithm is phrased as a single estimate for the data in question (described to the algorithm via the triple  $A, b, x_A$ , as opposed to Algorithm 1, which returned the entire list of  $m$  semilinear estimator coefficients). The following theorem, characterizing the performance of the above algorithm, immediately implies Theorem 1.

**Theorem 4.** *The mean squared error of the estimate output by Algorithm 2, over the randomness of the queried sample and target sets  $(A, b)$ , is within a multiplicative  $\frac{\pi}{2}$  factor and an additive  $6\epsilon$  factor of the performance of the optimum semilinear estimator, with probability  $1 - e^{-t \cdot \epsilon^5 / \text{poly}(n)}$  over the sampled inputs  $(A_{s_1}, b_{s_1}), \dots, (A_{s_t}, b_{s_t})$ . The probability of failure can thus be made exponentially small in  $n$  by using  $t = \text{poly}(n) / \epsilon^5$  samples, for a sufficiently large polynomial in  $n$ .*

We first prove three structural lemmas that characterize the optimization objective, and then put the pieces together making use of concentration bounds, applied over an  $\epsilon$ -net of matrices in the domain of the optimization.

**Lemma 1.** *For any valid  $V$ , the  $i^{\text{th}}$  term in the sum of Equation 3—or equivalently Equation 9 or Equation 10—is between 0 and 1.*

*Proof.* From the derivation of Equation 3 in the proof of Theorem 3, the inner summation is equal to the inner minimization in Equation 9, which we analyze instead. Since the quadratic form specified by  $V$  in Equation 9 is positive semidefinite, it thus always evaluates to a nonnegative number proving the first part of the claim.

Consider the inner minimum when all coefficients  $a_i$  are identically 0. Since each  $b_i$  is a nonnegative vector of sum 1, and thus since all entries of  $V$  have magnitude at most 1 (because of the diagonal constraint, and the positive semidefinite constraint), we have  $\sum_{j,k=1}^n b_{i(j)} b_{i(k)} V_{j,k} \leq 1$ , as desired.  $\square$

**Lemma 2.** *The optimum objective value of the max in Equation 3 decreases by at most  $\epsilon m$  if the domain of the maximization is further restricted so that  $V$ , instead of being positive semidefinite, must now have all eigenvalues at least  $\epsilon$ .*

*Proof.* From the derivation of Equation 3 in the proof of Theorem 3, the inner summation is equal to the inner minimization in Equation 9, which we analyze instead.

Letting  $V$  be the optimal matrix in Equation 9 we instead consider the matrix  $V_\epsilon = \epsilon I_n + (1 - \epsilon)V$  where  $I_n$  is the  $n \times n$  identity matrix. Since the objective is linear in  $V$ , when evaluated at  $V_\epsilon$  it will have value  $\epsilon$  times the objective value for  $I_n$ —which is nonnegative by Lemma 1—plus  $(1 - \epsilon)$  times its optimal objective value at  $V$ —which is at most 1 by Lemma 1. Thus  $V_\epsilon$  has objective value within  $\epsilon$  of the optimum, as desired.  $\square$

**Lemma 3.** *For a fixed symmetric matrix  $V$  whose eigenvalues are all at least  $\epsilon$ , the expression inside the sum of Equation 3, for any  $i$ , varies with respect to changing a coordinate of  $V$  by at most*

$$\left| \frac{d}{dV_{j,k}} b_i^T (V - V_{(A_i)}^T V_{(A_i, A_i)}^{-1} V_{(A_i)}) b_i \right| \leq \frac{1}{\epsilon^2} \text{poly}(n).$$

*Proof.* Since  $V$  has eigenvalues at least  $\epsilon$ , so does any (principal) submatrix  $V_{(A_i, A_i)}$ . Thus the inverse  $V_{(A_i, A_i)}^{-1}$  has eigenvalues at most  $\frac{1}{\epsilon}$ , and thus the  $L_2$  norm of any column of  $V_{(A_i, A_i)}^{-1}$  is at most  $\frac{1}{\epsilon}$ . Since  $\frac{d}{dV_{j,k}} V_{(A_i, A_i)}^{-1}$  equals negative the inner product of the columns (or rows)  $j$  and  $k$  of  $V_{(A_i, A_i)}^{-1}$ , this derivative is thus at most  $\frac{1}{\epsilon^2}$ . Applying the product rule can increase this by only a  $\text{poly}(n)$  factor.  $\square$

We assemble these pieces to prove the performance of Algorithm 2.

*Proof of Theorem 4.* For any fixed  $V$  in Equation 3, the average of the  $m$  terms in the sum may be estimated as the empirical average of the  $t$  terms we can compute from our randomly sampled inputs  $(A_{s_1}, b_{s_1}), \dots, (A_{s_t}, b_{s_t})$ . Since, by Lemma 1, each term is between 0 and 1, the standard Chernoff/Hoeffding bounds imply that the empirical mean of  $t$  random terms will be within  $\epsilon$  of the true mean except with probability  $e^{-2\epsilon^2 t}$ .

Let  $\epsilon' = \epsilon^3 / \text{poly}(n)$  be a radius such that, by Lemma 3, any two matrices satisfying the constraints of the arg max of Equation 10 that are within distance  $\epsilon'$  of each other must yield values for each term in the sum, that are within  $\epsilon$  of each other. Consider applying the concentration bounds of the previous paragraph to each  $V$  in an  $\epsilon'$ -net of matrices satisfying the conditions of Equation 10—namely, positive definite with eigenvalues at least  $\epsilon$ , and all diagonal entries at most 1. Recall that an  $\epsilon'$ -net will have each matrix within distance  $\epsilon'$  of one of the matrices in the net, and that the net will consist of  $e^{\text{poly}(n)/\epsilon'}$  matrices. As we consider bounds up to  $\text{poly}(n)$  factors, the choice of norm for the matrices does not matter, but for concreteness, consider the  $\epsilon'$ -net to be defined in the Frobenius norm. By the union bound, the Chernoff/Hoeffding bound of the previous paragraph applies for every  $V$  in the  $\epsilon'$ -net except with probability  $e^{-2\epsilon^2 t + \text{poly}(n)/\epsilon'}$ , which is thus negligible when the number of samples is  $t = \text{poly}(n)/\epsilon'\epsilon^2 = \text{poly}(n)/\epsilon^5$ .

We thus show that the performance of the estimator described by the sampled  $\tilde{V}$  is close to the performance of the optimal semilinear estimator  $\hat{V}$  with eigenvalues at least  $\epsilon$ . Let  $\tilde{V}', \hat{V}'$  respectively represent the nearest elements of the  $\epsilon'$ -net to  $\tilde{V}, \hat{V}$  respectively. For ease of notation, we let  $\hat{f}(V)$  and  $\tilde{f}(V)$  respectively describe the functions of  $V$  described by the average term in the sums of Equations 3 and 10 respectively. Thus we have

$$\hat{f}(\tilde{V}) \geq \hat{f}(\tilde{V}') - \epsilon \geq \tilde{f}(\tilde{V}') - 2\epsilon \geq \tilde{f}(\tilde{V}) - 3\epsilon \geq \tilde{f}(\hat{V}') - 3\epsilon \geq \hat{f}(\hat{V}') - 4\epsilon \geq \hat{f}(\hat{V}) - 5\epsilon,$$

where the inequalities hold respectively because of (1) the  $\epsilon'$ -nearness of  $\tilde{V}, \tilde{V}'$  combined with the derivative guarantee of Lemma 3 as applied to  $\hat{f}$ ; (2) the Chernoff/Hoeffding bound at the point  $\tilde{V}'$  of the  $\epsilon'$ -net; (3) the  $\epsilon'$ -nearness of  $\tilde{V}, \tilde{V}'$  combined with the derivative guarantee of Lemma 3 as applied to  $\tilde{f}$ ; (4) the fact that  $\tilde{V}$  attains the maximum of  $\tilde{f}$ ; (5) the Chernoff/Hoeffding bound at the point  $\tilde{V}'$  of the  $\epsilon'$ -net; and (6) the  $\epsilon'$ -nearness of  $\hat{V}, \hat{V}'$  combined with the derivative guarantee of Lemma 3 as applied to  $\hat{f}$ .

Thus, the algorithm described by  $\tilde{V}$  has true performance within  $5\epsilon$  of the optimal under the eigenvalue constraint, achieved by  $\hat{V}$ . By Lemma 2,  $\hat{V}$  itself is within  $\epsilon$  of the true optimal performance of Equation 3, which in turn is within a factor of  $\frac{\pi}{2}$  of that of the best semilinear estimator, as desired.  $\square$

## D Suboptimality of Semilinear Schemes (Fact 1)

Via a computer-aided brute-force search over small examples, we found a distribution,  $P$ , for which the best semilinear algorithm had larger worst-case expected error than the best arbitrary scheme:

**Example 4.** Let  $n = 4$ . Consider the distribution over sampling from the population  $\{1, 2, 3, 4\}$  that assigns a 0.3 probability to the following pairs of sample/target sets  $(\{1, 3\}, \{2, 4\})$ ,  $(\{2, 4\}, \{1, 3\})$ ,  $(\{3, 4\}, \{1, 2\})$  and a 0.05 probability to  $(\{1, 3, 4\}, \{2\})$  and  $(\{2, 3, 4\}, \{1\})$ . The optimal semilinear scheme achieves worst-case expected squared error 0.6652, compared to 0.6627 for the optimal unconstrained scheme. Hence even for sample/target set distributions over  $n = 4$  datapoints, semilinear schemes are not always worst-case optimal.

## E Proof of Theorem 2: Linear Regression Setting

We prove Theorem 2 here, which for clarity we restate and reintroduce slightly more formally.

We consider the following natural extension of our results to the setting of  $d$ -dimensional linear regression. Our regression results follow from a transparent application of the main results, Theorem 1 or Theorem 3, demonstrating the flexibility and scope of our approach. We emphasize that many variants of this model are interesting, and that a more specialized analysis may yield stronger bounds than what we show here.

**Definition 9.** Given a sample set  $A$  and a target set  $B$ , where for each  $i \in A$  we are given a pair  $(x_i, y_i)$  with the independent variable  $x_i \in [-1, 1]^d$  and the dependent variable  $y_i \in [-1, 1]$ , the goal is to recover a coefficient vector  $\beta$  that minimizes the mean squared error on the target set,  $\mathbb{E}_{i \sim B}[(x_i^\top \beta - y_i)^2]$ , when additionally given access to the joint distribution  $P$  from which  $(A, B)$  are drawn. The least-squares coefficients are defined, as is standard, as  $\beta = (E_{i \sim B}[x_i x_i^\top])^{-1} E_{i \sim B}[x_i y_i]$ . As we do not have access to the target set data  $(x_i, y_i)$  for  $i \in B$ , we instead must estimate it: depending on whether distribution  $P$  is given explicitly, or via sample access, use Theorems 1 or 3 respectively to estimate (in terms of  $P$  and the sample data  $(x_i, y_i)$  for  $i \in A$ ) each of the  $d^2$  entries in the matrix  $Q = E_{i \in B}[x_i x_i^\top]$  and the  $d$  entries in the vector  $u = E_{i \in B}[x_i y_i]$ , all of which are in  $[-1, 1]$ . Our estimated coefficients are then  $\hat{\beta} = \hat{Q}^{-1} \hat{u}$ .

**Theorem.** Given a regression problem as in Definition 9, where the distribution  $P$  is specified either via sampling access or explicitly, and let  $\alpha_P$  be the mean squared error guaranteed by Theorem 1 or 3 respectively, for estimating the mean of a scalar (which depends only on  $P$ ). Then, letting  $\sigma_d$  be the smallest singular value of the (uncentered covariance) matrix  $Q$ , for any  $\delta > 0$ , the algorithm of Definition 9 will return an estimate  $\hat{\beta}$  of the least-squares regression coefficients  $\beta$  such that with probability at least  $1 - \delta$  we have  $\|\beta - \hat{\beta}\| \leq 3 \frac{\sqrt{\alpha_P d^3 / \delta}}{\sigma_d^2}$ , provided this expression is at most 0.08; in the case that the independent variables  $x_i$  are known for the target set (and only the  $y_i$ 's are unknown) then except with  $\delta$  probability, we have  $\|\beta - \hat{\beta}\| \leq \frac{\sqrt{\alpha_P d / \delta}}{\sigma_d}$ .

*Proof.* Recall we are in the following regression setting: data consists of pairs  $(x_i, y_i)$  where  $x_i \in [-1, 1]^d$  and  $y_i \in [-1, 1]$ ; indices  $i \in [n]$  are drawn for the input sample  $A$  and target set  $B$  from a joint distribution,  $(A, B) \sim P$ . The goal, given the training data and a description of  $P$  (or sample access to  $P$ ), is to compute linear coefficients  $\beta \in \mathbb{R}^d$ , such that the mean squared error over target indices  $i \in B$  is as small as possible, namely, to minimize  $\mathbb{E}_{i \sim B}[(x_i^\top \beta - y_i)^2]$ . Our results will be parameterized in terms of  $\alpha_P$ , the mean squared estimation accuracy that Theorem 1 or Theorem 3 affords us on distribution  $P$  (in the scalar mean estimation setting).

As is standard in least squares regression, note that the expression we are minimizing,  $\mathbb{E}_{i \sim B}[(x_i^\top \beta - y_i)^2]$ , is positive semidefinite as a function of  $\beta$ , and thus is minimized when its gradient with respect to  $\beta$  equals 0. Hence we solve  $\mathbb{E}_{i \sim B}[x_i(x_i^\top \beta - y_i)] = 0$ , or equivalently,  $\mathbb{E}_{i \sim B}[x_i x_i^\top] \beta = \mathbb{E}_{i \sim B}[x_i y_i]$ , which has solution  $\beta = (E_{i \sim B}[x_i x_i^\top])^{-1} E_{i \sim B}[x_i y_i]$ . Now, for any  $i$ , each of the  $d^2$

entries of the matrix  $x_i x_i^\top$  is in  $[-1, 1]$ , and thus its average value over  $i$  in the target set  $B$  can be estimated to within mean squared error  $\alpha_P$  based on its values for  $i$  in the sample set  $A$ , from Theorem 1 or 3; the same holds for each of the  $d$  entries in the vector  $x_i y_i$ .

Let  $Q = \mathbb{E}_{i \sim B}[x_i x_i^\top]$ , and let  $Q + F$  (referred to as  $\widehat{Q}$  in definition 9) be the random variable representing the estimate of  $Q$  given by Theorem 1 or 3, where the square of each entry of  $F$  has expectation at most  $\alpha_P$ ; hence by Markov's inequality, for any  $\delta > 0$ , with probability at least  $1 - \delta$  the matrix  $F$  has Frobenius norm at most  $\sqrt{\alpha_P d^2 / \delta}$ . Correspondingly, let  $u = \mathbb{E}_{i \sim B}[x_i y_i]$ , and let  $u + g$  be the random variable representing the estimate of  $u$  returned by our algorithm, where, except with probability  $\delta$ , the vector  $g$  has length at most  $\sqrt{\alpha_P d / \delta}$ . Taking the union bound, we have that except with probability  $2\delta$ , the bounds on both  $F$  and  $g$  hold, and we analyze this case below.

As described above, the optimal linear coefficients are given by  $\beta = Q^{-1}u$ , and meanwhile our estimate is  $\hat{\beta} = (Q + F)^{-1}(u + g)$ . We bound the discrepancy  $\|\beta - \hat{\beta}\|$  via the triangle inequality, first bounding the change induced by adding  $g$ , and then bounding the change from adding  $F$ :

$$\|\beta - \hat{\beta}\| = \|Q^{-1}u - (Q + F)^{-1}(u + g)\| \leq \|Q^{-1}u - Q^{-1}(u + g)\| + \|Q^{-1}(u + g) - (Q + F)^{-1}(u + g)\|$$

The first term on the right hand side equals  $\|Q^{-1}g\|$ , which we bound as the product of the length of  $g$  and the largest singular value of  $Q^{-1}$ , which is the inverse of the smallest singular value of  $Q$ , which we have denoted  $\sigma_d$ . Thus we have  $\|Q^{-1}g\| \leq \frac{1}{\sigma_d} \sqrt{\alpha_P d / \delta}$ .

Bounding the second term is slightly more involved. The goal is to bound  $\|[(Q^{-1} - (Q + F)^{-1})(u + g)]\|$ . We first bound the largest singular value of the matrix  $Q^{-1} - (Q + F)^{-1}$ . For any  $\lambda \in [0, 1]$ , consider interpolating between  $Q$  and  $Q + F$  to get  $Q + \lambda F$ . Let  $s_\lambda$  be the smallest singular value of this matrix, and let  $v_\lambda$  be the corresponding singular vector, with  $\|v_\lambda\| = 1$ . Since  $Q$  has smallest singular value  $\sigma_d$ , we have  $\|Qv_\lambda\| \geq \sigma_d$ ; since  $F$  has Frobenius norm at most  $\sqrt{\alpha_P d^2 / \delta}$  and the Frobenius norm bounds the largest singular value, we have  $\|Fv_\lambda\| \leq \sqrt{\alpha_P d^2 / \delta}$ , and by the triangle inequality, the difference of these two expressions is a lower bound on  $s$ :  $s = \|(Q + \lambda F)v_\lambda\| \geq \sigma_d - \sqrt{\alpha_P d^2 / \delta}$ .

Consider the matrix  $(Q + \lambda F)^{-1}$  as we linearly move  $\lambda$  from 0 to 1. The derivative with respect to  $\lambda$  of this matrix inverse is  $-(Q^{-1} + \lambda F)F(Q + \lambda F)^{-1}$ , which thus has largest singular value at most the product of our bounds on the singular values for the 3 terms:

$$\frac{\sqrt{\alpha_P d^2 / \delta}}{(\sigma_d - \sqrt{\alpha_P d^2 / \delta})^2} \quad (11)$$

Since  $\|u + g\| \leq \sqrt{d} + \sqrt{\alpha_P d / \delta}$  from summing bounds on  $\|u\|$  and  $\|g\|$ , we multiply this by Equation 11—bounding the amount the matrix  $(Q + \lambda F)^{-1}$  changes as we interpolate from  $Q$  to  $Q + F$ —to get our total bound for the second triangle inequality term. Adding this to the bound on the first term, we have

$$\|\beta - \hat{\beta}\| = \|(Q + F)^{-1}(u + g) - Q^{-1}u\| \leq \frac{1}{\sigma_d} \sqrt{\alpha_P d / \delta} + (\sqrt{d} + \sqrt{\alpha_P d / \delta}) \frac{\sqrt{\alpha_P d^2 / \delta}}{(\sigma_d - \sqrt{\alpha_P d^2 / \delta})^2} \quad (12)$$

To simplify this bound, consider the case that  $\frac{\sqrt{\alpha_P d^3 / \delta}}{\sigma_d^2} \leq c$ .

Since matrix  $Q$  has entries in  $[-1, 1]$ , its singular values are at most  $\sqrt{d}$ , and thus  $\frac{\sqrt{d}}{\sigma_d} \geq 1$ , yielding  $\frac{\sqrt{\alpha_P d^2 / \delta}}{\sigma_d} \leq c$ , and implying that if we replace the denominator  $(\sigma_d - \sqrt{\alpha_P d^2 / \delta})^2$  in Equation 12 by simply  $\sigma_d^2$  then the expression will decrease by at most a factor of  $(1 - c)^2$ . Similarly, in Equation 12 the parenthetical term  $(\sqrt{d} + \sqrt{\alpha_P d / \delta})$  has second part bounded by  $\sqrt{\alpha_P d / \delta} \leq c \frac{\sigma_d}{\sqrt{d}} \leq c$  and thus replacing  $(\sqrt{d} + \sqrt{\alpha_P d / \delta})$  by simply  $\sqrt{d}$  will decrease the term by at most a factor of  $\frac{1}{1+c}$ . Thus the right hand side of Equation 12 is the sum of two terms,  $\frac{\sqrt{\alpha_P d / \delta}}{\sigma_d}$ , and a term  $\frac{\sqrt{\alpha_P d^3 / \delta}}{\sigma_d^2}$  times a factor between 1 and  $\frac{1+c}{(1-c)^2}$ ; the first term is clearly at most the second term since  $\sigma_d \leq \sqrt{d} \leq d^2$ , so thus we have  $\|\beta - \hat{\beta}\| \leq (1 + \frac{1+c}{(1-c)^2}) \frac{\sqrt{\alpha_P d^3 / \delta}}{\sigma_d^2}$ . Substituting  $2\delta \rightarrow \delta$  so that the overall probability

of failure becomes  $\delta$ , the proportionality constant becomes  $\sqrt{2}(1 + \frac{1+c}{(1-c)^2})$ , which for  $c \leq 0.0378$  yields a constant of 3, as desired. Thus, in the context of the theorem, substituting in  $\frac{1}{2}\delta$  for  $\delta$ , when  $3\frac{\sqrt{\alpha_P d^3 / (\delta/2)}}{\sigma_d^2} \leq 0.08$ , we will have  $c \leq \frac{0.08\sqrt{2}}{3} < 0.0378$  as desired.

In the simpler case where the independent variables  $x_i$  are known, and hence  $Q$  does not need to be estimated, our probability of failure is  $\delta$  instead of  $2\delta$ , and only the first term from Equation 12 appears, immediately yielding the other part of the theorem.

□