Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)

Online Active Linear Regression via Thresholding

Carlos Riquelme and **Ramesh Johari**
Stanford University,
{rikel, rjohari}@stanford.edu.

Baosen Zhang
Washington University,
zhangbao@uw.edu.

Abstract

We consider the problem of online active learning to collect data for regression modeling. Specifically, we consider a decision maker with a limited experimentation budget who must efficiently learn an underlying linear population model. Our main contribution is a novel threshold-based algorithm for selection of most informative observations; we characterize its performance and fundamental lower bounds. We extend the algorithm and its guarantees to sparse linear regression in high-dimensional settings. Simulations suggest the algorithm is remarkably robust: it provides significant benefits over passive random sampling in real-world datasets that exhibit high nonlinearity and high dimensionality — significantly reducing both the mean and variance of the squared error.

1 Introduction

This paper studies *online active learning* for estimation of linear models. Active learning is motivated by the premise that in many sequential data collection scenarios, labeling or obtaining output from observations is costly. Thus ongoing decisions must be made about whether to collect data on a particular unit of observation. Active learning has a rich history; see, e.g., (Cohn, Ghahramani, and Jordan 1996; Cohn, Atlas, and Ladner 1994; Castro and Nowak 2007; Koltchinskii 2010; Balcan, Hanneke, and Vaughan 2010).

As a motivating example, suppose that an online marketing organization plans to send display advertising promotions to a new target market. Their goal is to estimate the revenue that can be expected for an individual with a given covariate vector. Unfortunately, providing the promotion and collecting data on each individual is costly. Thus the goal of the marketing organization is to acquire first the most "informative" observations. They must do this in an online fashion: opportunities to display the promotion to individuals arrive sequentially over time. In online active learning, this is achieved by selecting those observational units (target individuals in this case) that provide the most information to the model fitting procedure.

Linear models are ubiquitous in both theory and practice—often used even in settings where the data may exhibit strong nonlinearity—in large part because of their interpretability, flexibility, and simplicity. As a consequence, in practice, people tend to add a large number of features and interactions to the model, hoping to capture the right signal at the expense of introducing some noise. Moreover, the input space can be updated and extended iteratively *after* data collection if the decision maker feels predictions on a held-out set are not good enough. As a consequence, often times the number of covariates becomes higher than the number of available observations. In those cases, selecting the subsequent most informative data is even more critical. Accordingly, our focus is on actively choosing observations for optimal *prediction* of the resulting high-dimensional linear models.

Our main contributions are as follows. We initially focus on standard linear models, and build the theory that we later extend to high dimensional settings. First, we develop an algorithm that sequentially selects observations if they have sufficiently large norm, in an appropriate space (dependent on the data-generating distribution). Second, we provide a comprehensive theoretical analysis of our algorithm, including upper and lower bounds. We focus on minimizing mean squared prediction error (MSE), and show a high probability upper bound on the MSE of our approach (cf. Theorem 3.1). In addition, we provide a lower bound on the best possible achievable performance in high probability and expectation (cf. Section 4). In some distributional settings of interest we show that this lower bound structurally matches our upper bound, suggesting our algorithm is near-optimal.

The results above show that the improvement of active learning progressively weakens as the dimension of the data grows, and a new approach is needed. To tackle our original goal and address this degradation, under standard sparsity assumptions, we design an adaptive extension of the thresholding algorithm that initially devotes some budget to learn the sparsity pattern of the model, in order to subsequently apply active learning to the relevant lower dimensional subspace. We find that in this setting, the active learning algorithm provides significant benefit over passive random sampling. Theoretical guarantees are given in Theorem 3.3.

Finally, we empirically evaluate our algorithm's performance. Our tests on real world data show our approach is remarkably robust: the gain of active learning remains significant even in settings that fall outside our theory. Our results suggest that the threshold-based rule may be a valuable tool to leverage in observation-limited environments, even when

the assumptions of our theory may not exactly hold.

Active learning has mainly been studied for classification; see, e.g., (Balcan, Beygelzimer, and Langford 2006; Dasgupta, Monteleoni, and Hsu 2007; Balcan, Broder, and Zhang 2007; Wang and Singh 2014; Dasgupta and Hsu 2008). For regression, see, e.g., (Krause and Guestrin 2007; Sugiyama and Nakajima 2009; Cai, Zhang, and Zhou 2013) and the references within. A closely related work to our setting is (Sabato and Munos 2014): they study online or stream-based active learning for linear regression, with random design. They propose a theoretical algorithm that partitions the space by stratification based on Monte-Carlo methods, where a recently proposed algorithm for linear regression (Hsu and Sabato 2014) is used as a black box. It converges to the globally optimal oracle risk under possibly misspecified models (with suitable assumptions). Due to the relatively weak model assumptions, they achieve a *constant* gain over passive learning. As we adopt stronger assumptions (well-specified model), we are able to achieve larger than constant gains, with a computationally simpler algorithm. Suppose covariate vectors are Gaussian with dimension d; the total number of observations is n; and the algorithm is allowed to label at most k of them. Then, we beat the standard $\sigma^2 d/k$ MSE to obtain $\sigma^2 d^2/[kd + 2(\delta - 1)k \log k]$ when $n = k^\delta$, so active learning truly improves performance when $k = \Omega(\exp(d))$ or $\delta = \Omega(d)$. While (Sabato and Munos 2014) does not tackle high-dimensional settings, we overcome the exponential data requirements via l_1-regularization.

The remainder of the paper is organized as follows. We define our setting in Section 2. In Section 3, we introduce the algorithm and provide analysis of a corresponding upper bound. Lower bounds are given in Section 4. Simulations are presented in Section 5, and Section 6 concludes.

2 Problem Definition

The online active learning problem for regression is defined as follows. We sequentially observe n covariate vectors in a d-dimensional space $X^i \in \mathbf{R}^d$, which are i.i.d. When presented with the i-th observation, we must choose whether we want to *label* it or not, i.e., choose to observe the outcome. If we decide to label the observation, then we obtain $Y^i \in \mathbf{R}$. Otherwise, we do not see its label, and the outcome remains unknown. We can label at most k out of the n observations.

We assume covariates are distributed according to some *known* distribution $\mathbf{D}$, with zero mean $\mathbf{E}X = 0$, and covariance matrix $\Sigma = \mathbf{E}XX^T$. We relax this assumption later. In addition, we assume that Y follows a linear model: $Y = X^T\beta^* + \epsilon$, where $\beta^* \in \mathbf{R}^d$ and $\epsilon \sim \mathcal{N}(0, \sigma^2)$ i.i.d. We denote observations by $X, X^i \in \mathbf{R}^d$, components by $X_j \in \mathbf{R}$, and sets in boldface: $\mathbf{X} \in \mathbf{R}^{k\times d}, \mathbf{Y} \in \mathbf{R}^k$.

After selecting k observations, $(\mathbf{X}, \mathbf{Y})$, we output an estimate $\hat{\beta}_k \in \mathbf{R}^d$, with no intercept.[1] Our goal is to minimize the expected MSE of $\hat{\beta}_k$ in Σ norm, i.e. $\mathbf{E}\|\hat{\beta}_k - \beta^*\|^2_\Sigma$, under random design; that is, when the X_i's are random and the algorithm may be randomized. This is related to the *A*-optimality criterion, (Pukelsheim 1993). We use the experimentation budget to minimize the variance of $\hat{\beta}_k$ by sampling $\mathbf{X}$ from a different thresholded distribution. Minimizing expected MSE is equivalent to minimizing the trace of the normalized inverse of the *Fisher information matrix* $\mathbf{X}^T\mathbf{X}$,

$$\begin{aligned}\mathbf{E}[(Y - X^T\hat{\beta}_k)^2] &= \mathbf{E}[\|\hat{\beta}_k - \beta^*\|^2_\Sigma] + \sigma^2 \\ &= \sigma^2\, \mathbf{E}\left[\mathrm{Tr}(\Sigma(\mathbf{X}^T\mathbf{X})^{-1})\right] + \sigma^2\end{aligned}$$

where expectations are over all sources of randomness. In this setting, the OLS estimator is the best linear unbiased estimator by the *Gauss–Markov Theorem*. Also, for any set $\mathbf{X}$ of k i.i.d. observations, $\hat{\beta}_k := \hat{\beta}^{OLS}_k$ has sampling distribution $\hat{\beta}_k \mid \mathbf{X} \sim \mathcal{N}(\beta^*, \sigma^2(\mathbf{X}^T\mathbf{X})^{-1})$, (Hoerl and Kennard 1970). In Section 3, we tackle high-dimensionality, where $k \le d$, via Lasso estimators within a two-stage algorithm.

3 Algorithm and Main Results

In this section we motivate the algorithm, state the main result quantifying its performance for general distributions, and provide a high-level overview of the proof. A corollary for the Gaussian distribution is presented, and we also extend the algorithm by making the threshold adaptive. Finally, we show how to generalize the results to *sparse* linear regression. In Appendix E[2], we derive a CLT approximation with guarantees that is useful in complex or unknown distributional settings.

Without loss of generality, we assume that each observation is *white*, that is, $\mathbf{E}[XX^T]$ is the identity matrix. For correlated observations X', we apply $X := D^{-1/2}U^TX'$ to whiten them, $\Sigma = UDU^T$ (see Appendix A). Note that $\mathrm{Tr}(\Sigma(\mathbf{X'}^T\mathbf{X'})^{-1}) = \mathrm{Tr}((\mathbf{X}^T\mathbf{X})^{-1})$.

We bound the whitened trace as

$$\frac{d}{\lambda_{\max}(\mathbf{X}^T\mathbf{X})} \le \mathrm{Tr}((\mathbf{X}^T\mathbf{X})^{-1}) \le \frac{d}{\lambda_{\min}(\mathbf{X}^T\mathbf{X})}. \quad (1)$$

To minimize the expected MSE, we need to maximize the minimum eigenvalue of $\mathbf{X}^T\mathbf{X}$ with high probability. The thresholding procedure in Algorithm 1 maximizes the minimum eigenvalue of $\mathbf{X}^T\mathbf{X}$ through two observations. *First*, since the sum of eigenvalues of $\mathbf{X}^T\mathbf{X}$ is the trace of $\mathbf{X}^T\mathbf{X}$, which is in turn the sum of the norm of the observations, the algorithm chooses observations of large (weighted) norm. *Second*, the eigenvalues of $\mathbf{X}^T\mathbf{X}$ should be balanced, that is, have similar magnitudes. This is achieved by selecting the appropriate weights for the norm.

Let $\xi \in \mathbf{R}^d_+$ be a vector of weights defining the norm $\|X\|^2_\xi = \sum_{j=1}^d \xi_j X_j^2$. Let $\Gamma > 0$ be a threshold. Algorithm 1 simply selects the observations with ξ-weighted norm larger than Γ. The selected observations can be thought as i.i.d. samples from an induced distribution $\bar{\mathbf{D}}$: the original distribution conditional on $\|X\|_\xi \ge \Gamma$. Suppose k observations are chosen and denoted by $\mathbf{X} \in \mathbf{R}^{k\times d}$. Then $\mathbf{E}\mathbf{X}^T\mathbf{X} = \sum_{i=1}^k \mathbf{E}X^i{X^i}^T = \sum_{i=1}^k H^i = kH$, where H is the covariance matrix with respect to $\bar{\mathbf{D}}$. This covariance

[1] We assume covariates and outcome are centered.

[2] The Appendix can be found in the Arxiv version of the paper.

matrix is diagonal under density symmetry assumptions, as thresholding preserves uncorrelation; its diagonal terms are

$$H_{jj} = \mathbf{E}_{\bar{\mathbf{D}}} X_j^2 = \mathbf{E}_{\mathbf{D}}[X_j^2 \mid \|X\|_\xi \geq \Gamma] =: \phi_j. \quad (2)$$

Hence, $\lambda_{\min}(\mathbf{E}\mathbf{X}^T\mathbf{X}) = k \min_j \phi_j$, and $\lambda_{\max}(\mathbf{E}\mathbf{X}^T\mathbf{X}) = k \max_j \phi_j$. The main technical result in Theorem 3.1 is to link the eigenvalues of the random matrix $\mathbf{X}^T\mathbf{X}$ to its deterministic counter part $\mathbf{E}\mathbf{X}^T\mathbf{X}$. From the above calculations, the goal is to find (ξ, Γ) such that $\min_j \phi_j \approx \max_j \phi_j$, and both are as large as possible. The first objective is achieved when there exists some ϕ such that

$$\mathbf{E}_{\mathbf{D}}[X_j^2 \mid \|X\|_\xi \geq \Gamma] = \phi_j = \phi, \text{ for all } j. \quad (3)$$

We note that if X has independent components with the same marginal distribution (after whitening), then it suffices to choose $\xi_j = 1$ for all j. It is necessary to choose unequal weights when the marginal distributions of the components are different, e.g., some are Gaussian and some are uniform, or components are dependent. For joint Gaussian, whitening removes dependencies, so we set $\xi_j = 1$.

Thresholding Algorithm

The algorithm is simple. For each incoming observation X^i we compute its weighted norm $\|X^i\|_\xi$ (possibly after whitening if necessary). If the norm is above the threshold Γ, then we select the observation, otherwise we ignore it. We stop when we have collected k observations. Note that *random sampling* is equivalent to setting $\Gamma = 0$.

We want to catch the k largest observations given our budget, therefore we require that Γ satisfies

$$\mathbf{P}_{\mathbf{D}}(\|X\|_\xi \geq \Gamma) = k/n. \quad (4)$$

If we apply this rule to n independent observations coming from $\mathbf{D}$, on average we select k of them: the $\xi-$largest. If (ξ, Γ) is a solution to (3) and (4), then $(c\,\xi, \sqrt{c}\,\Gamma)$ is also a solution for any $c > 0$. So we require $\sum_i \xi_i = d$.

Algorithm 1 Thresholding Algorithm.

Set $(\xi, \Gamma) \in \mathbf{R}^{d+1}$ satisfying (3) and (4).
Set $S = \emptyset$.
for observation $1 \leq i \leq n$ **do**
  Observe X^i.
  Compute $X^i = D^{-1/2}U^T X^i$.
  if $\|X^i\|_\xi > \Gamma$ or $k - |S| = n - i + 1$ **then**
    Choose X^i: $S = S \cup X^i$.
    if $|S| = k$ **then**
      break.
    end if
  end if
end for
Return OLS estimate $\hat{\beta}$ based on observations in S.

Algorithm 1 can be seen as a regularizing process similar to ridge regression, where the amount of regularization depends on the distribution $\mathbf{D}$ and the budget ratio k/n; it improves the conditioning of the problem.

Guarantees when Σ is unknown can be derived as follows: we allocate an initial sequence of points to estimation of the inverse of the covariance matrix, and the remainder to labeling (where we *no* longer update our estimate). In this manner observations remain independent. Note that $O(d)$ observations are required for accurate recovery when $\mathbf{D}$ is subgaussian, and $O(d \log d)$ if subexponential, (Vershynin 2010). Errors by using the estimate to whiten and make decisions are bounded, small with high probability (via Cauchy–Schwarz), and the result is equivalent to using a slightly worse threshold.

Algorithm 1 b Adaptive Thresholding Algorithm.

Set $S = \emptyset$.
for observation $1 \leq i \leq n$ **do**
  Observe X^i, estimate $\widehat{\Sigma}_i = \widehat{U}_i \widehat{D}_i \widehat{U}_i^T$.
  Compute $X^i = \widehat{D}_i^{-1/2} \widehat{U}_i^T X_i$.
  Let (ξ_i, Γ_i) satisfy (3) and (5).
  if $\|X^i\|_{\xi_i} > \Gamma_i$ or $k - |S| = n - i + 1$ **then**
    Choose X^i: $S = S \cup X^i$.
    if $|S| = k$ **then**
      break.
    end if
  end if
end for
Return OLS estimate $\hat{\beta}$ based on observations in S.

Algorithm 1 keeps the threshold fixed from the beginning, leading to a mathematically convenient analysis, as it generates i.i.d. observations. However, Algorithm 1b, which is adaptive and updates its parameters after each observation, produces slightly better results, as we empirically show in Appendix K. Before making a decision on X^i, Algorithm 1b finds (ξ_i, Γ_i) satisfying (3) and

$$\mathbf{P}_{\mathbf{D}}\left(\|X^i\|_{\xi_i} \geq \Gamma_i\right) = \frac{k - |S_{i-1}|}{n - i + 1}, \quad (5)$$

where $|S_{i-1}|$ is the number of observations already labeled. The idea is identical: set the threshold to capture, on average, the number of observations still to be labeled, that is $k - |S_{i-1}|$, out of the number still to be observed, $n - i + 1$.

Importantly, active learning not only decreases the expected MSE, but also its variance. Since the variance of the MSE for fixed $\mathbf{X}$ depends on $\sum_j 1/\lambda_j(\mathbf{X}^T\mathbf{X})^2$ (Hoerl and Kennard 1970), it is also minimized by selecting observations that lead to large eigenvalues of $\mathbf{X}^T\mathbf{X}$.

Main Theorem

Theorem 3.1 states that by sampling k observations from $\bar{\mathbf{D}}$ where (ξ, Γ) satisfy (3), the estimation performance is significantly improved, compared to randomly sampling k observations from the original distribution. Section 4 shows the gain in Theorem 3.1 essentially cannot be improved and Algorithm 1 is optimal. A sketch of the proof is provided at the end of this section (see Appendix B).

Theorem 3.1 *Let $n > k > d$. Assume observations $X \in \mathbf{R}^d$ are distributed according to subgaussian $\mathbf{D}$ with covariance matrix $\Sigma \in \mathbf{R}^{d \times d}$. Also, assume marginal densities are symmetric around zero after whitening. Let $\mathbf{X}$ be a $k \times d$ matrix with k observations sampled from the distribution induced*

by the thresholding rule with parameters $(\xi, \Gamma) \in \mathbf{R}_+^{d+1}$ *satisfying* (3). *Let* $\alpha > 0$, *so that* $t = \alpha\sqrt{k} - C\sqrt{d} > 0$, *then, with probability at least* $1 - 2\exp(-ct^2)$

$$\mathrm{Tr}(\Sigma(\mathbf{X}^T\mathbf{X})^{-1}) \leq \frac{d}{(1-\alpha)^2\,\phi k}, \tag{6}$$

where constants c, C *depend on the subgaussian norm of* $\bar{\mathbf{D}}$.

While Theorem 3.1 is stated in fairly general terms, we can apply the result to specific settings. We first present the Gaussian case where white components are independent. The proof is in Appendix D.

Corollary 3.2 *If the observations in Theorem 3.1 are jointly Gaussian with covariance matrix* $\Sigma \in \mathbf{R}^{d\times d}$, $\xi_j = 1$ *for all* $j = 1, \ldots, d$, *and* $\Gamma = \bar{C}\sqrt{d + 2\log(n/k)}$, *for some constant* $\bar{C} \geq 1$, *then with probability at least* $1 - 2\exp(-ct^2)$ *we have that*

$$\mathrm{Tr}(\Sigma(\mathbf{X}^T\mathbf{X})^{-1}) \leq \frac{d}{(1-\alpha)^2\left(1 + \frac{2\log(n/k)}{d}\right)\,k}. \tag{7}$$

The MSE of random sampling for white Gaussian data is proportional to $d/(k-d-1)$, by the inverse Wishart distribution. Active learning provides a gain factor of order $1/(1 + 2\log(n/k)/d)$ with high probability (a very similar $1-\alpha$ term shows up for random sampling). Note that our algorithm may select *fewer* than k observations. Then, when the number of observations yet to be seen equals the remaining labeling budget, we should select all of them (equivalent to random sampling). The number of observations with $\|X\|_\xi > \Gamma$ has binomial distribution, is highly concentrated around its mean k, with variance $k(1-k/n)$. By the Chernoff Bounds, the probability that the algorithm selects fewer than $k - C'\sqrt{k}$ decreases exponentially fast in C'. Thus, these deviations are dominated in the bound of Theorem 3.1 by the leading term. In practice, one may set the threshold in (4) by choosing $k(1+\epsilon)$ observations for some small $\epsilon > 0$, or use the adaptive threshold in Algorithm 1b.

Sparsity and Regularization

The gain provided by active learning in our setting suffers from the curse of dimensionality, as it diminishes very fast when d increases, and Section 4 shows the gain cannot be improved in general. For high dimensional settings (where $k \leq d$) we assume *s-sparsity* in β, that is, we assume the support of β contains at most s non-zero components, for some $s \ll d$. In Appendix J, we also provide related results for Ridge regression.

We state the two-stage *Sparse Thresholding* Algorithm (see Algorithm 2) and show this algorithm effectively overcomes the curse of dimensionality. For simplicity, we assume the data is Gaussian, $\mathbf{D} = \mathcal{N}(0, \Sigma)$. Based, for example, on the results of Tropp (2005) and Theorem 1 in Joseph (2013) we could extend our results to subgaussian data via the Orthogonal Matching Pursuit algorithm for recovery. The two-stage algorithm works as follows. First, we focus on recovering the true support, $S = S(\beta)$, by selecting the very first k_1 observations (without thresholding), and computing the Lasso estimator $\hat{\beta}_1$. Second, we assign the weights ξ: for $i \in S(\hat{\beta}_1)$, we set $\xi_i = 1$, otherwise we set $\xi_i = 0$. Then, we apply the thresholding rule to select the remaining $k_2 = k - k_1$ observations. While observations are collected in all dimensions, our final estimate $\hat{\beta}_2$ is the OLS estimator computed only including the observations selected in the second stage, and exclusively in those dimensions in $S(\hat{\beta}_1)$.

Note that, in general, the points that end up being selected by our algorithm are informational outliers, while not necessarily geometric outliers in the original space. After applying the whitening transformation, ignoring some dimensions based on the Lasso results, and then thresholding based on a weighted norm possibly learnt from data (say, if components are not independent, and we recover the covariance matrix in a online fashion), the algorithm is able to identify good points for the underlying data distribution and β.

Algorithm 2 Sparse Thresholding Algorithm.

Set $S_1 = \emptyset, S_2 = \emptyset$. Let $k = k_1 + k_2, n = k_1 + n_2$.
for observation $1 \leq i \leq k_1$ **do**
  Observe X^i. Choose X^i: $S_1 = S_1 \cup X^i$.
end for
Set $\gamma = 1/2, \lambda = \sqrt{4\sigma^2\log(d)/\gamma^2 k_1}$.
Compute Lasso estimate $\hat{\beta}_1$ on S_1, regularization λ.
Set weights: $\xi_i = 1$ if $i \in S(\hat{\beta}_1)$, $\xi_i = 0$ otherwise.
Set $\Gamma = C\sqrt{s + 2\log(n_2/k_2)}$.
Factorize $\Sigma_{S(\hat{\beta}_1)S(\hat{\beta}_1)} = UDU^T$.
for observation $k_1 + 1 \leq i \leq n$ **do**
  Observe $X^i \in \mathbf{R}^d$. Restrict to $X^i_S := X^i_{S(\hat{\beta}_1)} \in \mathbf{R}^s$.
  Compute ${X^i}_S = D^{-1/2}U^T X^i_S$.
  if $\|X^i_S\|_\xi > \Gamma$ or $k_2 - |S_2| = n - i + 1$ **then**
    Choose X^i_S: $S_2 = S_2 \cup X^i_S$.
    if $|S_2| = k_2$ **then**
      break.
    end if
  end if
end for
Return OLS estimate $\hat{\beta}_2$ based on observations in S_2.

Theorem 3.3 summarizes the performance of Algorithm 2; it requires the standard assumptions on Σ, λ and $\min_i |\beta_i|$ for support recovery (see Theorem 3 in (Wainwright 2009)).

Theorem 3.3 *Let* $\mathbf{D} = \mathcal{N}(0, \Sigma)$. *Assume* Σ, λ *and* $\min_i |\beta_i|$ *satisfy the standard conditions given in Theorem 3 of (Wainwright 2009). Assume we run the Sparse Thresholding algorithm with* $k_1 = C's\log d$ *observations to recover the support of* β, *for an appropriate* $C' \geq 0$. *Let* $\mathbf{X}_2$ *be* $k_2 = k - k_1$ *observations sampled via thresholding on* $S(\hat{\beta}_1)$. *It follows that for* $\alpha > 0$ *such that* $t = \alpha\sqrt{k_2} - C\sqrt{s} > 0$, *there exist some universal constants* c_1, c_2, *and* c, C *that depend on the subgaussian norm of* $\bar{\mathbf{D}} \mid S(\hat{\beta}_1)$, *such that with probability at least*

$$1 - 2e^{-\min\left(c_2\min(s,\log(d-s)) - \log(c_1), ct^2 - \log(2)\right)}$$

it holds that

$$\mathrm{Tr}(\Sigma_{SS}(\mathbf{X}_2^T\mathbf{X}_2)^{-1}) \leq \frac{s}{(1-\alpha)^2\left(1+\frac{2\log(n_2/k_2)}{s}\right)\,k_2}.$$

Performance for random sampling with the Lasso estimator is $O(s\log d/k)$. A regime of interest is $s \ll d, k = C_1 s\log d$, and $n = C_2\, d$, for large enough C_1, and $C_2 > 0$. In that case, Algorithm 2 leads to a bound of order smaller than $1/\log(d)$, as opposed to a weaker constant guarantee for random sampling. The gain is at least a $\log d$ factor with high probability. The proof is in Appendix H. In practice, the performance of the algorithm is improved by using all the k observations to fit the final estimate $\hat{\beta}_2$. However, in that case, observations are no longer i.i.d. Also, using thresholding to select the initial k_1 observations decreases the probability of making a mistake in support recovery. In Section 5 we provide simulations comparing different methods.

Proof of Theorem 3.1

The complete proof of Theorem 3.1 is in Appendix B. We only provide a sketch here. The proof is a direct application of spectral results in (Vershynin 2010), which are derived via a covering argument using a discrete net $\mathcal{N}$ on the unit Euclidean sphere S^{d-1}, together with a Bernstein-type concentration inequality that controls deviations of $\|\mathbf{X}w\|_2$ for each element $w \in \mathcal{N}$ in the net. Finally, a union bound is taken over the net. Importantly, the proof shows that if our algorithm uses (ξ, Γ) which are *approximate* solutions to (3), then (6) still holds with $\min_j \mathbf{E}_{\bar{\mathbf{D}}} X_j^2$ in the denominator of the RHS, instead of ϕ. This fact can be quite useful in practice, when $\mathbf{F}$ is unknown. We can devote some initial budget $X_1, \ldots, X_T$ to recover $\mathbf{F}$, and then find (ξ, Γ) approximately solving (3) and (4) under $\hat{\mathbf{F}}$. Note that no labeling is required.

Also, the result can be extended to subexponential distributions. In this case, the probabilistic bound will be weaker (including a d term in front of the exponential). More generally, our probabilistic bounds are strongest when $k \geq Cd\log d$ for some constant $C \geq 0$, a common situation in active learning (Sabato and Munos 2014), where super-linear requirements in d seem unavoidable in noisy settings. A simple bound for the parameter ϕ can be calculated as follows. Assume there exists (ξ, Γ) such that $\phi_j = \phi$ and consider the weighted squared norm $Z_\xi = \sum_{j=1}^d \xi_j X_j^2$. Then $\mathbf{E}_{\bar{D}}\left[Z_\xi\right] = \sum_{j=1}^d \xi_j \mathbf{E}_{\bar{D}}\left[X_j^2\right] = \sum_{j=1}^d \xi_j\phi_j = d\phi$, and $\phi = \mathbf{E}_D\left[Z_\xi \mid Z_\xi \geq \Gamma^2\right]/d \geq \Gamma^2/d = F_{Z_\xi}^{-1}(1-k/n)/d$, which implies that $1/\lambda_{\min}(\mathbf{E}\mathbf{X}^T\mathbf{X}) = 1/k\phi \leq d/k\Gamma^2$. For specific distributions, Γ^2/d can be easily computed. The last inequality is close to equality in cases where the conditional density decays extremely fast for values of $\sum_{j=1}^d \xi_j X_j^2$ above Γ^2. Heavy-tailed distributions allocate mass to significantly higher values, and ϕ could be much larger than Γ^2/d.

4 Lower Bound

In this section we derive a lower bound for the $k > d$ setting. Suppose all the data are given. Again choose the k observations with largest norms, denoted by $\mathbf{X}'$. To minimize the prediction error, the best possible ${\mathbf{X}'}^T\mathbf{X}'$ is diagonal, with identical entries, and trace equal to the sum of the norms. No selection algorithm, online or offline, can do better. Algorithm 1 achieves this by selecting observations with large norms and uncorrelated entries (through whitening if necessary). Theorem 4.1 captures this intuition.

Theorem 4.1 *Let* $\mathbf{A}$ *be an algorithm for the problem we described in Section 2. Then,*

$$\mathbf{E}_{\mathbf{A}}\,\mathrm{Tr}(\Sigma(\mathbf{X}^T\mathbf{X})^{-1}) \geq \frac{d^2}{\mathbf{E}\left[\sum_{i=1}^k ||X_{(i)}||^2\right]} \geq \frac{d}{k\,\mathbf{E}\left[\frac{1}{d}\,\max_{i\in[n]}||X_i||^2\right]}, \tag{8}$$

where $X_{(i)}$ *is the white observation with the* i*-th largest norm. Moreover, fix* $\alpha \in (0,1)$*. Let* $\mathbf{F}$ *be the cdf of* $\max_{i\in[n]}||X_i||^2$*. Then,* $\mathrm{Tr}(\Sigma(\mathbf{X}^T\mathbf{X})^{-1}) \geq d^2/k\,\mathbf{F}^{-1}(1-\alpha)$ *with probability at least* $1-\alpha$*.*

The proof is in Appendix E. The upper bound in Theorem 3.1 has a similar structure, with denominator equal to $k\phi$. By Theorem 3.1, $\phi = \mathbf{E}_{\mathbf{D}}[X_j^2 \mid \|X\|_\xi^2 \geq \Gamma^2]$ for every component j. Hence, summing over all components: $k\phi = k\mathbf{E}_{\bar{\mathbf{D}}}\left[\|X\|^2/d\right]$. The latter expectation is taken with respect to $\bar{\mathbf{D}}$, which only captures the k expected ξ-largest observations out of n, as opposed to $k\,\mathbf{E}_{\mathbf{D}}[(1/k)\sum_{i=1}^k ||X_{(i)}||^2/d]$ in (8). The weights ξ simply account for the fact that, in reality, we cannot make all components have equal norm, something we implicitly assumed in our lower bound.

We specialize the lower bound to the Gaussian setting, for which we computed the upper bound of Theorem 3.1. The proofs are based on the Fisher-Tippett Theorem and the Gumbel distribution; see Appendix F.

Corollary 4.2 *For Gaussian observations* $X^i \sim \mathcal{N}(0,\Sigma)$ *and large* n*, for any algorithm* $\mathbf{A}$

$$\mathbf{E}_{\mathbf{A}}\,\mathrm{Tr}(\Sigma(\mathbf{X}^T\mathbf{X})^{-1}) \geq \frac{d}{k\left(\frac{2\log n}{d}+\log\log n\right)}.$$

Moreover, let $\alpha \in (0,1)$*. Then, for any* $\mathbf{A}$ *with probability at least* $1-\alpha$ *and* $C = 2\log\Gamma(d/2)/d$*,*

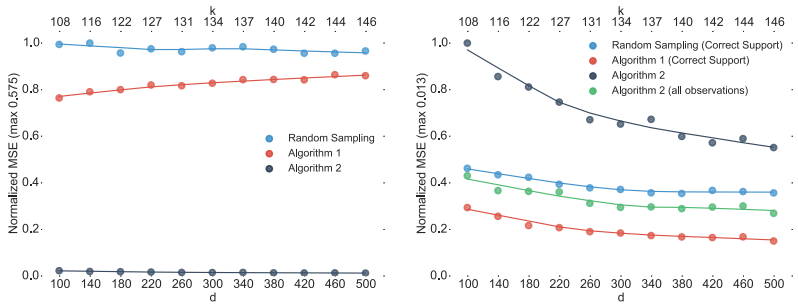
$$\mathrm{Tr}(\Sigma(\mathbf{X}^T\mathbf{X})^{-1}) \geq \frac{d/k}{\frac{2\log n}{d}+\log\log n-\frac{1}{d}\log\log\frac{1}{1-\alpha}-C}$$

The results from Corollary 3.2 have the same structure as the lower bound; hence in this setting our algorithm is near optimal. Similar results and conclusions are derived for the CLT approximation in Appendix I.

5 Simulations

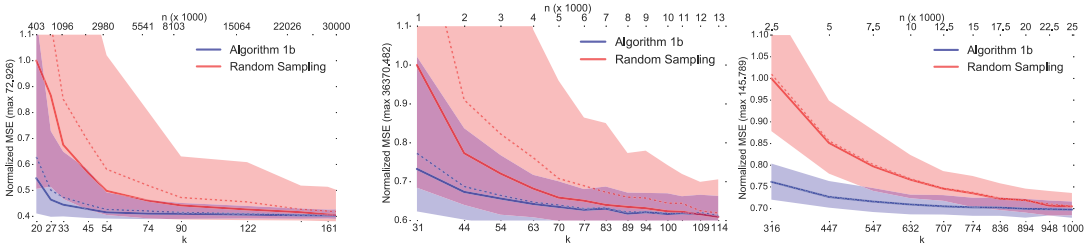
We conducted experiments in various settings: regularized estimators in high-dimensions, and the basic thresholding approach in real-world data to explore its performance on strongly non-linear environments.

Regularized Estimators. We compare the performance in high-dimensional settings of random sampling and Algorithm 1 —both with an appropriately adjusted Lasso estimator—



(a) Zooming out. (b) Zooming in.

Figure 1: Sparse Linear Regression (700 iters). We fix the effective dimension to $s = 7$, and increase the ambient dimension from $d = 100$ to $d = 500$. The budget scales as $k = Cs \log d$ for $C \approx 3.4$, while $n = 4d$. We set $k_1 = 2k/3$ and $k_2 = k/3$.



(a) Protein Structure; 150 iters. (b) Bike Sharing; 300 iters. (c) YearPredictionMSD; 150 iters

Figure 2: MSE of $\hat{\beta}_{OLS}$. The $(0.05, 0.95)$ quantile conf. int. displayed. Solid *median*; Dashed *mean*.

against Algorithm 2, which takes into account the structure of the problem ($s \ll d$). For completeness, we also show the performance of Algorithm 2 when *all* observations are included in the final OLS estimate, and that of random sampling (RS) and Algorithm 1 (Thr) when the true support S is known in advance, and the OLS computed on S. In Figure 1 (a), we see that Algorithm 2 dramatically reduces the MSE, while in Figure 1 (b) we zoom-in to see that, quite remarkably, Algorithm 2 using all observations for the final estimate outperforms random sampling that knows the sparsity pattern in hindsight. We used $k_1 = (2/3)k$ for recovery. More experiments are provided in Appendix K.

Real-World Data. We show the results of Algorithm 1b (online Σ estimation) with the simplest distributional assumption (Gaussian threshold, $\xi_j = 1$) versus random sampling on publicly available real-world datasets (UCI, (Lichman 2013)), measuring test squared prediction error. We fix a sequence of values of n, together with $k = \sqrt{n}$, and for each pair (n, k) we run a number of iterations. In each one, we randomly split the dataset in training (n observations, random order), and test (rest of them). Finally, $\hat{\beta}_{\mathrm{OLS}}$ is computed on selected observations, and the prediction error estimated on the test set. All datasets are initially centered to have zero means (covariates and response). Confidence intervals are provided.

We first analyze the Physicochemical Properties of Protein Tertiary Structure dataset (45730 observations), where we predict the size of the residue, based on $d = 9$ variables, including the total surface area of the protein and its molecular mass. Figure 2 (a) shows the results; Algorithm 1b outperforms random sampling for all values of (n, k). The reduction in variance is substantial. In the Bike Sharing dataset (Fanaee-T and Gama 2013) we predict the number of hourly users of the service, given weather conditions, including temperature, wind speed, humidity, and temporal covariates. There are 17379 observations, and we use $d = 12$ covariates. Our estimator has lower mean, median and variance MSE than random sampling; Figure 2 (b). Finally, for the YearPredictionMSD dataset (Bertin-Mahieux et al. 2011), we predict the year a song was released based on $d = 90$ covariates, mainly metadata and audio features. There are 99799 observations. The MSE and variance did strongly improve; Figure 2 (c).

In the examples we see that, while active learning leads to strong improvements in MSE and variance reduction for moderate values of k with respect to d, the gain vanishes when k grows large. This was expected; the reason might be that by sampling so many outliers, we end up learning about parts of the space where heavy non-linearities arise, which may not be important to the test distribution. However, the motivation of active learning are situations of limited labeling budget, and hybrid approaches combining random sampling and thresholding could be easily implemented if needed.

6 Conclusion

Our paper provides a comprehensive analysis of thresholding algorithms for online active learning of linear regression models, which are shown to perform well both theoretically and empirically. Several natural open directions suggest themselves. Additional robustness could be guaranteed in other

settings by combining our algorithm as a "black box" with other approaches: for example, some addition of random sampling or stratified sampling could be used to determine if significant nonlinearity is present, and to determine the fraction of observations that are collected via thresholding.

7 Acknowledgments

The authors would like to thank Sven Schmit for his excellent comments and suggestions, Mohammad Ghavamzadeh for fruitful discussions, and the anonymous reviewers for their valuable feedback. We gratefully acknowledge support from the National Science Foundation under grants CMMI-1234955, CNS-1343253, and CNS-1544548.

References

Balcan, M.-F.; Beygelzimer, A.; and Langford, J. 2006. Agnostic active learning. In *Proceedings of the 23rd international conference on Machine learning*, 65–72. ACM.

Balcan, M.-F.; Broder, A.; and Zhang, T. 2007. Margin based active learning. In *Learning Theory*. Springer. 35–50.

Balcan, M.-F.; Hanneke, S.; and Vaughan, J. W. 2010. The true sample complexity of active learning. *Machine learning* 80(2-3):111–139.

Bertin-Mahieux, T.; Ellis, D. P.; Whitman, B.; and Lamere, P. 2011. The million song dataset.

Cai, W.; Zhang, Y.; and Zhou, J. 2013. Maximizing expected model change for active learning in regression. In *Data Mining (ICDM), 2013 IEEE 13th International Conference on*, 51–60. IEEE.

Castro, R. M., and Nowak, R. D. 2007. Minimax bounds for active learning. 5–19.

Cohn, D.; Atlas, L.; and Ladner, R. 1994. Improving generalization with active learning. *Machine learning* 15(2):201–221.

Cohn, D. A.; Ghahramani, Z.; and Jordan, M. I. 1996. Active learning with statistical models. *Journal of artificial intelligence research*.

Dasgupta, S., and Hsu, D. 2008. Hierarchical sampling for active learning. In *Proceedings of the 25th international conference on Machine learning*, 208–215. ACM.

Dasgupta, S.; Monteleoni, C.; and Hsu, D. J. 2007. A general agnostic active learning algorithm. In *Advances in neural information processing systems*, 353–360.

Fanaee-T, H., and Gama, J. 2013. Event labeling combining ensemble detectors and background knowledge. *Progress in Artificial Intelligence* 1–15.

Hoerl, A. E., and Kennard, R. W. 1970. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics* 12(1):55–67.

Hsu, D., and Sabato, S. 2014. Heavy-tailed regression with a generalized median-of-means. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, 37–45.

Koltchinskii, V. 2010. Rademacher complexities and bounding the excess risk in active learning. *The Journal of Machine Learning Research* 11:2457–2485.

Krause, A., and Guestrin, C. 2007. Nonmyopic active learning of gaussian processes: an exploration-exploitation approach. In *Proceedings of the 24th international conference on Machine learning*, 449–456. ACM.

Lichman, M. 2013. UCI machine learning repository.

Pukelsheim, F. 1993. *Optimal design of experiments*, volume 50. siam.

Sabato, S., and Munos, R. 2014. Active regression by stratification. In *Advances in Neural Information Processing Systems*, 469–477.

Sugiyama, M., and Nakajima, S. 2009. Pool-based active learning in approximate linear regression. *Machine Learning* 75(3):249–274.

Vershynin, R. 2010. Introduction to the non-asymptotic analysis of random matrices. *arXiv preprint arXiv:1011.3027*.

Wainwright, M. J. 2009. Sharp thresholds for high-dimensional and noisy sparsity recovery using-constrained quadratic programming (lasso). *Information Theory, IEEE Transactions on* 55(5):2183–2202.

Wang, Y., and Singh, A. 2014. Noise-adaptive margin-based active learning and lower bounds under tsybakov noise condition. *arXiv preprint arXiv:1406.5383*.