Active Linear Regression in the Online Setting via Leverage Score Sampling

Harvineet Singh and **Christopher Musco** and **Rumi Chunara**
New York University

Abstract

We study an online version of active learning for linear function fitting. Given a stream of data points, the goal is to maintain an accurate linear regression solution while querying as few responses as possible *online* – i.e., without the possibility to revisit points after they appear in the stream. Building on techniques from randomized numerical linear algebra, we propose and analyze a practical procedure that samples responses with probability proportional to each data point's *leverage score*. The procedure gives a guaranteed approximation to the full-sample regression solution, notably, without any assumptions on the data distribution. Although extensive experiments reveal that the procedure does not perform uniformly well across real datasets, we propose two variants that show stronger empirical performance.

Keywords: active learning, linear regression, leverage scores, online learning

1. Introduction

We study the following question – given a stream of feature vectors arriving one at a time, how can we approximate the least squares linear regression solution for the entire stream by only querying responses for a subset of the feature vectors? In many applications, the full dataset is not available prior to analysis and arrives in an online manner. Moreover, querying responses corresponding to the features might be costly, thus, prompting the data analyst to limit the number of queries. As an example, consider a stream of patients visiting a hospital with certain symptoms (features). The health status (response) is only observed by conducting expensive tests. How can we reduce the number of responses queried while maintaining a pre-determined level of error for the trained predictor?

Formally, we consider the active regression problem in an online setting. Figure 1 summarizes the setting. Suppose we receive rows from a design matrix $A \in \mathbb{R}^{n\times d}$ in a stream $a_1, a_2, ..., a_n$ where $a_i \in \mathbb{R}^d$ is the i^{th} row. For each a_i, we can sample its response $b_i \in \mathbb{R}$ from the response vector $b \in \mathbb{R}^n$. Note that we are allowed to observe and store the rows of A but only selectively sample elements of b. Our goal is to find a vector $x \in \mathbb{R}^d$ using the collected samples that approximates the minimum achievable squared error $\min_x \|Ax - b\|_2^2$.

We refer the reader to Chen et al. (2021) for a summary of active learning settings considered in prior work. We depart from existing work on the active linear regression problem in two ways. First, we are interested in approximation error at any given time point in the stream. Most work in online learning, either in the classification (Atlas et al., 1989; Cesa-Bianchi et al., 2009; Dekel et al., 2012) or regression settings (Vovk, 1997; Azoury and Warmuth, 2000), measures performance in terms of *cumulative* error, i.e. sum of approximation errors over time steps. Applications such as the healthcare example described earlier typically separate the data collection period from deployment. The model is put into practice only after the data collection period. Hence, the error at the end of the data

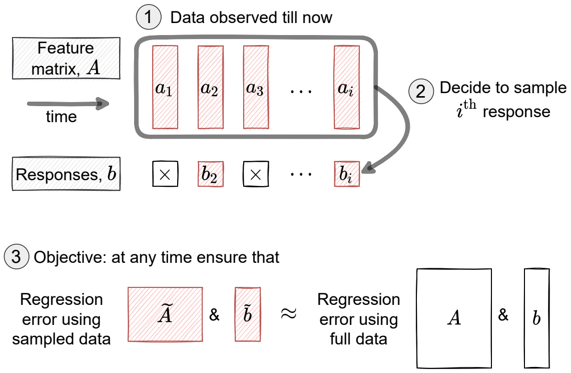


Figure 1: **Overview.** In online active regression, we observe a stream of feature vectors (1) and use them to decide whether to sample the current response (2). Objective is to find a linear regression solution from the sampled features and responses that achieves error close to the one found if we could observe the full stream (3).

collection period is of interest rather than the cumulative error throughout the period. A 'single time' performance measure for active linear regression has been considered in prior work (e.g. in (Drineas et al., 2006; Derezinski et al., 2018; Sabato and Munos, 2014; Chen and Price, 2019)) but prior work does not consider a streaming setting – instead, all data points are assumed to be available when the responses are queried actively. The streaming setting in Calandriello et al. (2017) also differs from ours since it assumes that both the responses and the data points are provided at each time point. Their goal is to lower the computation cost by keeping fewer data points, as opposed to reducing the number of responses queried.

Secondly, existing work typically makes assumptions on the data distribution for analyzing error rates of the algorithms (Riquelme et al., 2017; Cavallanti et al., 2008; Cesa-Bianchi et al., 2009). The closest work to ours is Riquelme et al. (2017) which has the same online active linear regression setting but assumes that the data points are sampled independently from some distribution and the response follows an additive Gaussian noise model. Our problem setting and the performance measure is framed in an adversarial setting and accordingly the analysis does not require making distributional assumptions.

Our approach to the problem is based on importance sampling, that is, we subsample features and responses with some probability, and subsequently reweight them while solving the least squares problem (Drineas et al., 2006). We show that a simple sampling-and-reweighting method based on *leverage scores* adapts to the online setting and guarantees an approximate solution. In addition to the analysis, we add to the sparse literature on experimentally evaluating leverage scores for regression problems on multiple realistic datasets (Ma et al., 2015, 2020; Orhan and Tastan, 2018).

2. Methodology

Consider a sequence of probabilities $p_1, p_2, ..., p_n$ which we will use to sample elements of b. Say, $S \in \mathbb{R}^{n \times n}$ is a diagonal sampling matrix where each diagonal element S_{ii} is $1/\sqrt{p_i}$ if we sample at step i, otherwise S_{ii} is 0. Then, at the end of the stream we obtain $\tilde{x} = \text{argmin}_x \|SAx - Sb\|_2^2$. Our goal will be to ensure that, with high probability, the squared error of $\tilde{x}$ for the original least squares objective $\|A\tilde{x} - b\|_2^2$ is at most a multiplicative factor worse than the optimal objective value $\min_x \|Ax - b\|_2^2$. To construct the sampling matrix, we will use *online leverage scores* devised by Cohen et al. (2016) for obtaining a spectral approximation of a matrix.

Online Leverage Scores Let A_i denotes the matrix observed till step i of the stream. The online leverage score for row a_i is defined as $\ell_i = \min(a_i^T(A_i^T A_i)^{-1} a_i, 1)$. In theory, we add an identity matrix with a small constant to avoid non-invertible matrices $a_i^T(A_i^T A_i + \lambda I)^{-1} a_i$, known as λ-ridge leverage scores. In practice, we use the pseudo-inverse $(A_i^T A_i)^{+}$. We sample the response with probability $p_i = \min(c\ell_i, 1)$ where c is a positive constant that controls the sampling rate. We reweight by $1/\sqrt{p_i}$ to maintain unbiased expectation of the test error. Finally, we solve the least squares problem on the sampled data. Algorithm 1 outlines the procedure. The key observation made by Cohen et al. (2016), which is sufficient for our main result, is that online leverage scores *overestimate* the true leverage scores (that is, leverage scores computed for the full matrix A as in the offline setting).

```
Set $c = k\log(d)/\epsilon^2$ for a fixed constant $k$.
Initialize $\tilde{A}_0 \leftarrow [\,], \tilde{b}_0 \leftarrow [\,]$.
for $i = 1, \ldots, n$ rows do
    Observe $a_i$.
    Compute online leverage score $\ell_i = a_i^T(A_i^T A_i)^{+} a_i$.
    Compute sampling probability $p_i = \min(c\; l_i, 1)$.
    Set $\tilde{A}_i, \tilde{b}_i \leftarrow \begin{cases} \begin{bmatrix} \tilde{A}_{i-1} \\ a_i/\sqrt{p_i} \end{bmatrix}, \begin{bmatrix} \tilde{b}_{i-1} \\ b_i/\sqrt{p_i} \end{bmatrix} & \text{with probability } p_i, \\ \tilde{A}_{i-1}, \tilde{b}_{i-1}, & \text{otherwise.} \end{cases}$
end
Result: $\tilde{x}^* = \mathrm{argmin}_x \|\tilde{A}x - \tilde{b}\|_2^2$.
```

Algorithm 1: $\tilde{x}^* = \text{LEVERAGE}(A, b, \epsilon, \delta)$, where A is the $n \times d$ data matrix, b is $n \times 1$ response vector, $\epsilon, \delta \in (0, 1)$, and a tuning parameter k.

2.1 Our Main Result

We state an upper bound for the number of samples needed for an ϵ-multiplicative error approximation in the online setting. We contrast it to the offline setting where we have access to A and b beforehand. Here, the condition number of A is denoted by $\kappa(A) := \|A\|_2 \|A^{-1}\|_2$.

Theorem 1 (Online Least Squares Guarantee) *Suppose we design the sampling matrix S with online λ_{min}-ridge leverage score sampling, where $\lambda_{min} := \lambda_{min}(A^T A)$ is the minimum eigenvalue of $A^T A$. Let $\tilde{x}^* = argmin_x \|SAx - Sb\|_2^2$. Then with $O\left(d \log d \log(1+\kappa^2(A)) + \frac{d}{\delta\epsilon}\log(1+\kappa^2(A))\right)$ samples, the following holds with probability at least $(1-\delta)$,*

$$\|A\tilde{x}^* - b\|_2^2 \leq (1+\epsilon)\|Ax^* - b\|_2^2, \tag{1}$$

where $x^* = argmin_x \|Ax - b\|_2^2$.

For the offline setting, results by Drineas et al. (2006) and subsequent improvements guarantee that leverage score sampling with $O\left(d \log d + \frac{d}{\delta\epsilon}\right)$ samples is sufficient for approximate least squares regression. Thus, online leverage score sampling at most adds a multiplicative factor $\log(1+\kappa^2(A))$ to the sampling cost which depends on the given A. We prove the main result in Section B that involves showing that online leverage scores ensure spectral approximation (Section A.1) and approximate matrix multiplication (Section A.2).

2.2 Another Method - Online Root Leverage Score Sampling

For the *offline* active regression problem, Ma et al. (2020) propose to sample by square root of the true leverage scores. Their work motivates this sampling method through an analysis of the statistical properties of the resulting estimators in a linear model with zero-mean noise. Root leverage score sampling achieves the best asymptotic mean squared error among unbiased estimators in the sample-then-reweight category of estimators. We adapt this method to the online setting by computing *online* root leverage scores from the rows seen till any step i as $\ell_i = \min((a_i^T(A_i^TA_i)^{-1}a_i)^{1/2}, 1)$. We sample by these probabilities keeping other steps in the Algorithm 1 the same. We also implement a biased variant of leverage score sampling that follows the same sampling procedure but fits an *unweighted* least squares on the sampled points. That is, the step of reweighting by $1/\sqrt{p_i}$ in the Algorithm 1 is skipped. The intuition for unweighted regression is that we may reduce the high variance that results from sampling points with small probabilities (making $1/\sqrt{p_i}$ high) while incurring, possibly, small bias in the regression error because of no reweighting.

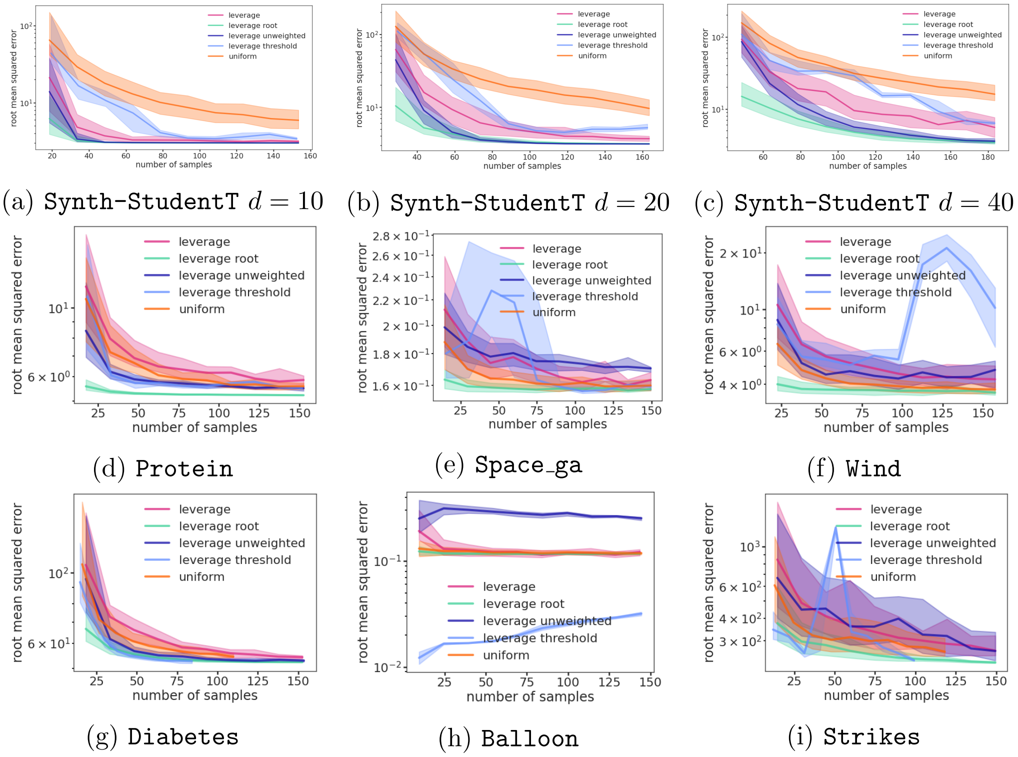
3. Experiments

The goal of the experiments is to check how the error rate of the two proposed sampling methods compare to the two baselines. *Uniform random sampling* samples each data point with a fixed probability determined before observing any data. Past work across different active learning settings remarks the effectiveness of uniform sampling on real datasets (Orhan and Tastan, 2018). *Leverage threshold* is a method introduced in (Riquelme et al., 2017) which queries a data point if its online leverage score exceeds an adaptively set threshold. We use the same dynamic thresholding Algorithm 1b in Riquelme et al. (2017) with Gaussian thresholds and ξ_i=1 as used in their experiments.

Practical implementation Our method requires a ridge parameter to prevent the non-invertability of $A_i^TA_i$ in the calculating the online leverage scores. We addressed this in theory by adding a small $\lambda_{\min} \cdot I$ matrix where $\lambda_{\min} := \lambda_{\min}(A^TA)$. However, $\lambda_{\min}$ is only known after seeing the whole stream A. As an alternative, we use the leverage scores defined with a pseudo-inverse which also prevents non-invertability. We compute leverage scores as $\tilde{\ell}_i = a_i^T(A_i^TA_i)^+a_i$, which is well-defined for any i. Further, we note that $\tilde{\ell}_i$ are overestimates of true $\lambda_{\min}$-ridge leverage scores. So, using the pseudo-inverse also guarantees online spectral approximation and matrix multiplication by Lemmas 2 and 4, and thus, the least squares approximation by Theorem 1 – the only possible disadvantage of using the pseudo-inverse is that we might collect more responses than guaranteed by Theorem 1.

Datasets In total we use 33 datasets. Following the setting T_1 in Ma et al. (2015), 3 are synthetically generated so that A has highly non-uniform leverage scores. The 30 real datasets are taken from OpenML using the query `https://www.openml.org/search?q=tags.tag%3Astudy_130%2520qualities.NumberOfMissingValues%3A0&type=data`. We include those that have greater than 500 examples, non-symbolic features, and less than 70 features. Links to individual datasets are in Table 1 and summary statistics are in Table 2 in Appendix C.

Evaluation setup For each dataset, we leave aside 30% of data points for testing. We traverse the training data points in their original order, thus, simulating a stream. The number of sampled outcomes depend on the hyperparameter ϵ (including k) which has to be tuned for the methods separately. As an alternative, we run the methods with various values of ϵ and compare them at the same number of sampled points. We repeatedly run each value of ϵ for 10 times. At any given point in the stream, we report root mean squared error (RMSE) achieved by the least squares solution on the left out 30% test dataset. Note that test error is a different measure than the error on the full matrix A considered in Theorem 1.



(a) `Synth-StudentT` $d = 10$ (b) `Synth-StudentT` $d = 20$ (c) `Synth-StudentT` $d = 40$

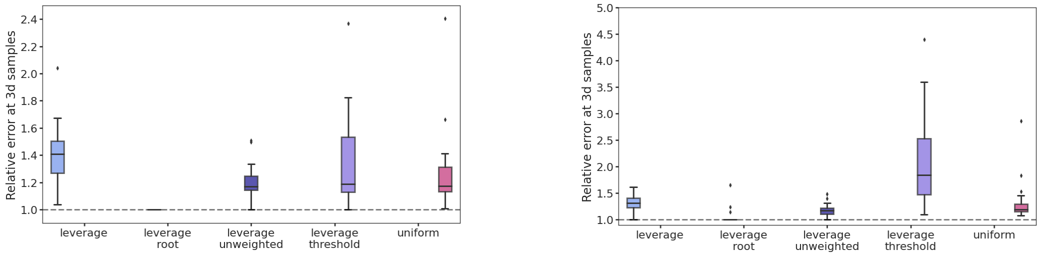
(d) `Protein` (e) `Space_ga` (f) `Wind`

(g) `Diabetes` (h) `Balloon` (i) `Strikes`

Figure 2: RMSE (in log scale) vs number of samples for different variants of leverage score sampling. We observe that online root leverage score sampling (leverage root, in cyan) requires less samples for the same error, thus, outperforming other methods substantially. Methods named 'leverage threshold' (Riquelme et al., 2017) and 'uniform' are baselines.

3.1 Results

Proposed method outperforms existing ones in sample complexity in synthetic and real world datasets Figures 2 shows the RMSE for the methods as they sample more rows. Figures 2a, 2b, and 2c report results for the synthetic setting with increasing number of features $d \in \{10, 20, 40\}$. Leverage score based methods consistently outperform uniform sampling (Uniform) since synthetic data has a few 'outlier' rows with high leverage scores that are important to sample to get low error. However in the real datasets, Uniform outperforms online leverage score sampling in some cases. On the other hand, online root leverage does better in most cases. Figures 2g, 2h, and 2i are some where the threshold-based baseline does better. Figure 3a summarizes the results across datasets by plotting the error *relative* to the best error at $3 \cdot d$ sampled data points, as achieved by any of the 5 methods. So, the best value is 1.0 and higher is worse. We observe that the relative error for leverage root is around 1.0 (median) and is better than the baselines. We will study the reasons for the variation in performance across datasets as part of further work.



(a) Original data order

(b) Adversarial data order

Figure 3: **Aggregate results from 30 datasets.** Box plot for each method's error relative to the best performing method after sampling $3 \cdot d$ points. Value for the best method will be 1.0. Methods named 'leverage threshold' (Riquelme et al., 2017) and 'uniform' are baselines. (a) Data points in the stream are presented in the original order of the points. (b) Data points are sorted in decreasing order of true leverage scores (aggr. on 29 datasets). In both (a) and (b), we observe that online root leverage score sampling (leverage root) achieves the best error (close to 1.0) relative to other leverage score variants and uniform sampling.

Improvements decrease as number of dimensions increase Results for the synthetic settings in Figures 2a, 2b, and 2c show that the gap in performance between Uniform and the rest of the methods gets smaller as d increases. This illustrates the sample size dependence seen in Theorem 1. Analysis of dependence on n, d for leverage root requires further work.

Methods are robust to adversarial ordering of data streams We test reliance on distributional assumptions on the stream. Figure 3b plots error vs number of samples when the data points in the stream are presented in the decreasing order of their true leverage scores. This ordering of the stream is adversarial to the assumptions of the thresholding-based baseline since it sets the threshold by assuming that the points are sampled independently from some distribution. Consequently, we see that leverage threshold does worse. Relative performance of the rest of the methods is the same as in the original data ordering.

4. Conclusion and Discussion

We studied online leverage score sampling for approximately solving least squares regression when rows of the design matrix and response arrive in a stream. This work extends the analysis of leverage scores from the randomized numerical algebra literature (Drineas et al., 2006; Cohen et al., 2016, 2015) to online linear regression. Using the observation from this literature that online leverage scores overestimate the true leverage scores, we prove that sampling by these scores guarantee a relative error approximation without needing assumptions on the data distribution. We show that solving least squares in the online setting adds at most a multiplicative factor (log of matrix's condition number) to the sampling cost of leverage score sampling in the offline setting. Further work should try to understand the strong performance of root leverage scores in the asymptotic or finite sample regimes. Another interesting direction is to consider the relation of root leverage score sampling to linear UCB methods (Auer et al., 2002) which also, in part, use the same scores.

Appendix A. Intermediate Results

A.1 Online Leverage Score Sampling for Spectral Approximation

The key observation made by Cohen et al. (2016) (see Theorem 2.1) is that the online leverage scores overestimate the standard leverage scores defined as $a_i^T(A^TA)^{-1}a_i$. Further, sampling with overestimates of leverage scores suffices to guarantee spectral approximation. Consider the following known result.

Lemma 2 (Lemma 4 in (Cohen et al., 2015)) *Given $A \in \mathbb{R}^{n\times d}$ and overestimates of leverage scores $o_i \geq a_i^T(A^TA)^{-1}a_i$ for all $i = 1, 2, ..., n$ and the desired error parameter $\epsilon > 0$. Let constant $c = \epsilon^{-2}\log d$. For $i = 1, \ldots, n$, let the sampling probability for row a_i is $p_i = min(co_i, 1)$. Define the diagonal sampling matrix $S \in \mathbb{R}^{n\times n}$ with $S_{ii} = 1/\sqrt{p_i}$ if the row is sampled and 0 otherwise. Then, with $O(c\,(\sum_{i=1}^n o_i))$ samples, the following holds with high probability,*

$$(1-\epsilon)A^TA \preceq A^TS^TSA \preceq (1+\epsilon)A^TA. \tag{2}$$

Suppose we set λ to be the minimum eigenvalue of A^TA, $\lambda = \lambda_{\min}(A^TA)$, then observe that the online λ-ridge leverage scores, adjusted by a constant, are overestimates i.e. $\ell_i = a_i^T(A_i^TA_i + \lambda_{\min}(A^TA)I)^{-1}a_i \geq \frac{1}{2}a_i^T(A^TA)^{-1}a_i$. Since $A_i^TA_i \preceq A^TA$, it follows that $A_i^TA_i + \lambda_{\min}(A^TA)I \preceq A^TA + \lambda_{\min}(A^TA)I \preceq 2A^TA$. Thereby, implying the above overestimate. We can adjust the factor of $1/2$ in the constant ϵ when determining the sampling probability p_i. Thus, Lemma 2 implies that the number of samples required for spectral approximation are of the order $O(\epsilon^{-2}\log d\,(\sum_{i=1}^n \ell_i))$.

Next, we restate a known bound on the sum of online leverage scores $\sum_{i=1}^n \ell_i$.

Lemma 3 (Sum of Online Leverage Scores, Theorem 2.2 in (Cohen et al., 2016)) *Given a matrix $A \in \mathbb{R}^{n\times d}$, let A_i denote the first i rows of A. Let the online λ-ridge leverage score for row a_i is $\ell_i = min(a_i^T(A_{i-1}^TA_{i-1} + \lambda I)^{-1}a_i, 1)$. Then $\sum_{i=1}^n \ell_i = O(d\log(1 + \|A\|_2^2/\lambda))$.*

Setting $\lambda = \lambda_{\min}(A^TA) = 1/\|A^{-1}\|_2^2$, the sum is $O(d\log(1 + (\|A\|_2^2\|A^{-1}\|_2^2))) = O(d\log(1 + \kappa^2(A)))$ for the condition number $\kappa(A) := \|A\|_2\|A^{-1}\|_2$. Combining this with Lemma 2, we have that $O(\epsilon^{-2}d\log d\log(1+\kappa^2(A)))$ samples guarantee spectral approximation. The result also holds for online leverage scores computed after including the i^{th} point, that is, using $A_i^TA_i$ in place of $A_{i-1}^TA_{i-1}$ in ℓ_i as done in Algorithm 1.

A.2 Online Leverage Score Sampling for Matrix Multiplication

Next, we derive a new result for approximate matrix multiplication with online leverage scores. Consider two matrices $W \in \mathbb{R}^{n\times d}$ and $Z \in \mathbb{R}^{n\times m}$ which we want to multiply. Let w_i denote the i^{th} row of W.

Lemma 4 (Online Matrix Multiplication) *Given a matrix $W \in \mathbb{R}^{n\times d}$ and a sequence of scores o_i defined such that $o_i \geq \|w_i\|_2^2/\|W\|_F^2$ for all $i = 1, 2, ..., n$ and the desired error parameter $\epsilon > 0$. Let constants $\delta > 0$ and $c = \epsilon^{-2}/\delta$. For $i = 1, \ldots, n$, let the sampling probability for row w_i is $p_i = min(co_i, 1)$. Define the diagonal sampling matrix $S \in \mathbb{R}^{n\times n}$*

with $S_{ii} = 1/\sqrt{p_i}$ if the row is sampled and 0 *otherwise. If we sample $O(c\sum_{i=1}^{n} o_i)$, then with probability $(1-\delta)$,*

$$\|W^T S^T S Z - W^T Z\|_F \leq \epsilon \|W\|_F \|Z\|_F.$$

Proof We will first show that the random matrix $W^T S^T S Z$ is an unbiased estimator of $W^T Z$. Then, we will use the Markov's inequality to show the required result. The arguments follow the proof for the offline case given in (Drineas and Mahoney, 2018) (see Theorem 22).

If we assume $\mathbb{E}[W^T S^T S Z] = W^T Z$ for now, then Markov's inequality implies,

$$\Pr\left[\|W^T S^T S Z - W^T Z\|_F > \epsilon \|W\|_F \|Z\|_F\right] < \frac{\mathbb{E}\|W^T S^T S Z - W^T Z\|_F^2}{\epsilon^2 \|W\|_F^2 \|Z\|_F^2}. \tag{3}$$

Thus, we first prove unbiasedness and then bound $\mathbb{E}\|W^T S^T S Z - W^T Z\|_F^2$. We will rewrite each element $(W^T S^T S Z)_{i,j}$ for some fixed i, j as a sum of rank-one matrices. For $t = 1, ..., n$, define $X_t = \frac{s_t}{p_t}(w_t z_t^T)_{i,j}$ where w_t and z_t are t^{th} rows of W and Z respectively. The row w_t is sampled with probability p_t and s_t is the binary indicator which is 1 when we sample and is 0 otherwise. Observe that $(W^T S^T S Z)_{i,j} = \sum_{t=1}^{n} X_t$. Then,

$$\mathbb{E}[(W^T S^T S Z)_{i,j}] = \mathbb{E}\left[\sum_t X_t\right] = \sum_t \mathbb{E}[X_t] = \sum_t p_t \left(\frac{1}{p_t}(w_t z_t^T)_{i,j}\right) = (W^T Z)_{i,j}$$

Thus, $\mathbb{E}[W^T S^T S Z] = W^T Z$.

Note that here we assumed that there is no randomness associated with W and Z, i.e. the matrices are fixed before the stream is started. Thus, sampling decision s_t is the only random variable in X_t. Moreover $s_1, ..., s_n$ are independent as the sampling probabilities depend on fixed matrices $W_0, ..., W_{n-1}$. It follows that $X_1, ..., X_n$ are independent random variables. Consider the variance of $(W^T S^T S Z)_{i,j}$ which can be written as follows

$$\mathrm{Var}[(W^T S^T S Z)_{i,j}] = \mathrm{Var}\left[\sum_t X_t\right] = \sum_t \mathrm{Var}[X_t] \leq \sum_t \mathbb{E}\left[X_t^2\right] = \sum_t \frac{1}{p_t}(w_{t,i}^2 z_{t,j}^2) \tag{4}$$

Next, we use this to bound $\mathbb{E}\|W^T S^T S Z - W^T Z\|_F^2$. Since $\mathbb{E}[W^T S^T S Z] = W^T Z$, we have that

$$\mathbb{E}\|W^T S^T S Z - W^T Z\|_F^2 = \sum_{i,j} \mathbb{E}(W^T S^T S Z - W^T Z)_{i,j}^2 = \sum_{i,j} \mathrm{Var}[(W^T S^T S Z)_{i,j}]$$

From (4), we get that

$$\mathbb{E}\|W^T S^T S Z - W^T Z\|_F^2 \leq \sum_{i,j} \sum_t \frac{1}{p_t}(w_{t,i}^2 z_{t,j}^2) = \sum_t \frac{1}{p_t}\|w_t\|_2^2 \|z_t\|_2^2 \tag{5}$$

First, we assume that all $p_i = c \cdot o_i < 1$ and return to the case when some p_i can be 1 later. By our assumption that $o_i \geq \|w_t\|^2 / \|W\|_F^2$, we get,

$$\sum_t \frac{1}{p_t}\|w_t\|_2^2 \|z_t\|_2^2 \leq \sum_t \frac{1}{c\|w_t\|^2/\|W\|_F^2}\|w_t\|_2^2 \|z_t\|_2^2 = \frac{1}{c}\|W\|_F^2 \sum_t \|z_t\|_2^2 = \frac{1}{c}\|W\|_F^2 \|Z\|_F^2$$

Then, substituting in (5), we have

$$\mathbb{E}\|W^T S^T SZ - W^T Z\|_F^2 \leq \frac{1}{c}\|W\|_F^2\|Z\|_F^2 \tag{6}$$

Putting this back in (3), we rewrite the failure probability as

$$\Pr\left[\|W^T S^T SZ - W^T Z\|_F > \epsilon\|W\|_F\|Z\|_F\right] < \frac{\|W\|_F^2\|Z\|_F^2}{c\epsilon^2\|W\|_F^2\|Z\|_F^2} = \frac{1}{c\epsilon^2}.$$

Now we come to general case when some $p_i = 1$. We observe that the rows sampled with probability 1 cancel out in the difference $W^T S^T SZ - W^T Z$ that we are interested in. Suppose we take all the rows sampled with probability 1 and put them in a matrix W'. Let the matrix Z' has the corresponding rows from Z. Define $W'' = W \setminus W'$ and $Z'' = Z \setminus Z'$ as matrices with the rest of the rows. Similarly split the sampling matrix into S', which has all entries equal to 1 as corresponding $p_i = 1$, and $S'' = S \setminus S'$. Then, by definition, $W^T Z = W'^T Z' + W''^T Z''$ and

$$W^T S^T SZ - W^T Z = W'^T Z' + W''^T S''^T S'' Z'' - (W'^T Z' + W''^T Z'') = W''^T S''^T S'' Z'' - W''^T Z''.$$

We get the same form as the expression in (5). Proceeding as earlier we recover the result in (6) since

$$\mathbb{E}\|W^T S^T SZ - W^T Z\|_F^2 = \mathbb{E}\|W''^T S''^T S'' Z'' - W''^T Z''\|_F^2 \leq \frac{1}{c}\|W''\|_F^2\|Z''\|_F^2 \leq \frac{1}{c}\|W\|_F^2\|Z\|_F^2$$

Therefore, we get the same failure probability in the general case $p_i \leq 1$,

$$\Pr\left[\|W^T S^T SZ - W^T Z\|_F > \epsilon\|W\|_F\|Z\|_F\right] < \frac{1}{c\epsilon^2}.$$

Thus, we set $\delta = \frac{1}{c\epsilon^2}$. Finally, the number of samples $\sum_t s_t$ is given by a Chernoff bound as $\sum_t p_t$ with high probability. From the definition of p_t, we know that $\sum_t p_t \leq \sum_t co_t$. Thus the number of samples $O(c(\sum_{t=1}^n o_t)))$. ■

In the proof of Theorem 1, we apply Lemma 3 again to bound $\sum_i o_i$ in the above result where o_i are online $\lambda_{\min}$-ridge leverage scores.

Appendix B. Main Result for Online Least Squares Approximation

Now, we will prove Theorem 1 using the preceding results. Note that the following proof largely follows the proof for the offline setting.

Theorem 1 (Online Least Squares Guarantee) *Suppose we design the sampling matrix S with online λ_{min}-ridge leverage score sampling, where $\lambda_{min} := \lambda_{min}(A^TA)$ is the minimum eigenvalue of A^TA. Let $\tilde{x}^* = argmin_x\|SAx - Sb\|_2^2$. Then with $O\left(d\log d\log(1+\kappa^2(A)) + \frac{d}{\delta\epsilon}\log(1+\kappa^2(A))\right)$ samples, the following holds with probability at least $(1-\delta)$,*

$$\|A\tilde{x}^* - b\|_2^2 \leq (1+\epsilon)\|Ax^* - b\|_2^2,$$

where $x^* = argmin_x\|Ax - b\|_2^2$.

Proof The arguments are based on the proof by Woodruff (2014) (see Theorem 2.16) modified for online leverage score sampling.

Note that $(Ax^* - b)$ is orthogonal to any vector in the column space of A. Since x^* is the minimizer, thus, the gradient $\nabla(\|Ax^* - b\|_2^2) = 2A^T(Ax^* - b)$ is equal to $\vec{0}$ at x^*. Specifically, $(Ax^* - b)$ is orthogonal to $A\tilde{x}^* - Ax^*$. By Pythagorean theorem, we have

$$\|A\tilde{x}^* - b\|_2^2 = \|Ax^* - b\|_2^2 + \|A\tilde{x}^* - Ax^*\|_2^2 \tag{7}$$

Next, we reparameterize A in terms of an orthogonal matrix. Let $U \in \mathbb{R}^{n\times d}$ be a matrix with orthonormal columns which spans column space of A. Therefore, $A\tilde{x}^* = U\tilde{y}^*$ for some $\tilde{y}^*$ and $Ax^* = Uy^*$ for some y^*. By this transformation, we have that (7) is equivalent to

$$\|U\tilde{y}^* - b\|_2^2 = \|Uy^* - b\|_2^2 + \|U\tilde{y}^* - Uy^*\|_2^2. \tag{8}$$

As orthogonal matrices preserve norms, thus, $\|U\tilde{y}^* - Uy^*\|_2^2 = \|\tilde{y}^* - y^*\|_2^2$. Therefore to bound the squared error, we will show that

$$\|\tilde{y}^* - y^*\|_2^2 \leq \epsilon\|Uy^* - b\|_2^2. \tag{9}$$

Since we sample A with online $\lambda_{\min}$-ridge leverage scores, we have from Lemma 3 that $\frac{1}{2}A^TA \preceq A^TS^TSA \preceq (1+\frac{1}{2})A^TA$ as long as we sample $O(d\log d\log(1+\kappa^2(A)))$ rows. Since for any x, we can find a y such that $Ax = Uy$, it follows that $\frac{1}{2}U^TU \preceq U^TS^TSU \preceq (1+\frac{1}{2})U^TU$. But $U^TU = I$, so this implies $\|U^TS^TSU - I\|_2 \leq \frac{1}{2}$. Then,

$$\begin{aligned}
\|\tilde{y}^* - y^*\|_2 &\leq \|U^TS^TSU(\tilde{y}^* - y^*)\|_2 + \|U^TS^TSU(\tilde{y}^* - y^*) - (\tilde{y}^* - y^*)\|_2 \\
&\leq \|U^TS^TSU(\tilde{y}^* - y^*)\|_2 + \|U^TS^TSU - I\|_2\|(\tilde{y}^* - y^*)\|_2 \\
&\leq \|U^TS^TSU(\tilde{y}^* - y^*)\|_2 + \frac{1}{2}\|\tilde{y}^* - y^*\|_2.
\end{aligned} \tag{10}$$

where the second inequality follows from sub-multiplicativity of the spectral norm. It follows from (10) that $\|\tilde{y}^* - y^*\|_2 \leq 2\|U^TS^TSU(\tilde{y}^* - y^*)\|_2$. We now focus on $\|U^TS^TSU(\tilde{y}^* - y^*)\|_2$ to prove (9).

Since $\tilde{y}^* = \text{argmin}_y\|SUy - Sb\|_2^2$, $SU\tilde{y}^* - Sb$ must be orthogonal to any vector in the column space of SU. It follows that

$$\|U^TS^TSU(\tilde{y}^* - y^*)\|_2 = \|U^TS^T(SU\tilde{y}^* - Sb + Sb - SUy^*)\|_2 = \|U^TS^TS(b - Uy^*)\|_2$$

Now we can apply Lemma 4. To do so we note that online $\lambda_{\min}$-ridge leverage scores of A are (up to a constant) overestimates of leverage scores of A which are equal to the ones for the orthogonal matrix U. To see this, $\ell_i = a_i^T(A_{i-1}^TA_{i-1} + \lambda_{\min}(A^TA)I)^{-1}a_i \geq 1/2a_i^T(A^TAI)^{-1}a_i = 1/2\|u_i\|_2^2$. Since $\|U\|_F^2 = d$, we have that the scores $\ell_i \geq d/2\|u_i\|_2^2/\|U\|_F^2$ satisfy the condition required by Lemma 4. After adjusting the factor $d/2$ and replacing $c = \epsilon^{-2}/\delta$, the number of samples needed is equivalent to $O(\epsilon^{-2}/\delta\log(1+\kappa^2(A)))$.

So if we sample $O(\frac{d}{\delta\epsilon}\log(1+\kappa^2(A)))$ rows by online leverage score sampling then

$$\|U^TS^TS(b - Uy^*)\|_2 \leq \frac{\sqrt{\epsilon}}{\sqrt{d}}\|U\|_F\|Uy^* - b\|_2.$$

with probability $(1-\delta)$. But $\|U\|_F = \sqrt{d}$, therefore, we get $\|U^TS^TSU(\tilde{y}^* - y)\|_2^2 \leq \epsilon\|Uy^* - b\|_2^2$.

Substituting in (10) gives $\|\tilde{y}^* - y^*\|_2^2 \leq 4\epsilon\|Uy^* - b\|_2^2$, which proves (9) after adjusting ϵ by a constant factor. ■

Appendix C. Dataset details

Table 1 lists the website links with details on how to access the datasets. Table 2 show some summary statistics of the dataset.

Name	URL
Synth-StudentT	Setting T_1 in Ma et al. (2015)
Diabetes	`https://scikit-learn.org/stable/datasets/toy_dataset.html#diabetes-dataset`
Magic04	`http://manikvarma.org/code/LDKL/download.html`
Mpg	`https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/regression.html`
CPUsmall	`https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/regression.html`
Abalone	`https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/regression.html`
California-Housing	`https://scikit-learn.org/stable/datasets/real_world.html#california-housing-dataset`
Cod-RNA	`https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary.html`
Communities-and-Crime	`https://github.com/slundberg/shap/tree/master/data`
Cover-Type	`https://archive.ics.uci.edu/ml/datasets/covertype`
Protein	`https://archive.ics.uci.edu/ml/datasets/Physicochemical+Properties+of+Protein+Tertiary+Structure`
BikeSharing	`https://archive.ics.uci.edu/ml/datasets/bike+sharing+dataset`
Concrete	`http://archive.ics.uci.edu/ml/datasets/concrete+compressive+strength`
PM10	`http://lib.stat.cmu.edu/datasets/`
Mg-scale	`https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/regression.html`
2dplanes,elevators,bank32nh, house_16H,wind,space_ga,houses, no2,strikes,balloon,puma32H, wine_quality,quake,stock,kin8nm,fried	`https://www.openml.org/search?q=tags.tag%3Astudy_130%2520qualities.NumberOfMissingValues%3A0&type=data`

Table 1: Details for accessing datasets.

Appendix D. Related Work

We discuss closely related work from two domains.

dataset	n	d	r2	log_cond	$\frac{\text{max lev}}{\text{median lev}}$
Synth-StudentT_d10	3500	11	1.000	5.660	3285.863
Synth-StudentT_d20	3500	21	1.000	7.533	3174.324
Synth-StudentT_d40	3500	41	1.000	6.579	1990.298
california_housing	5000	9	0.596	8.475	961.609
house_16H	5000	17	0.269	16.311	443.197
protein	5000	10	0.272	17.027	441.642
abalone	2924	9	0.551	4.413	291.491
cpusmall	5000	13	0.719	12.983	103.798
wine_quality	4548	12	0.300	11.265	91.637
houses	5000	9	0.635	8.385	63.319
covtype	3500	55	0.311	83.902	61.996
balloon	1401	2	0.674	1.765	57.846
magic	5000	11	0.319	2.663	37.454
communitiesandcrime	1396	102	0.688	43.795	17.476
elevators	5000	19	0.811	56.219	16.440
space_ga	2175	7	0.386	18.694	12.012
quake	1525	4	0.002	4.769	11.451
cod	3500	9	0.646	9.689	9.590
wind	4602	15	0.758	2.657	8.599
mpg	275	8	0.823	7.465	8.573
diabetes	310	11	0.512	5.534	7.353
strikes	438	7	0.085	3.854	6.973
concrete	721	9	0.604	4.741	6.409
bikesharing	5000	13	0.419	36.590	5.602
no2	350	8	0.513	5.596	4.826
pm10	350	8	0.157	5.652	3.892
bank32nh	5000	33	0.524	2.716	3.681
stock	665	10	0.970	3.176	3.306
kin8nm	5000	9	0.434	0.125	2.013
fried	5000	11	0.719	1.278	1.974
mg_scale	970	7	0.576	1.371	1.974
puma32H	5000	33	0.218	4.258	1.624
2dplanes	5000	11	0.712	0.243	1.486

Table 2: **Properties of datasets.** Total number of data points n, features d, R^2 coefficient from full-sample linear regression r2, logarithm of the condition number of the feature matrix log_cond are reported for each dataset. We sort the datasets in decreasing order of the ratio of maximum to median leverage scores $\frac{\text{max lev}}{\text{median lev}}$ that measures skewness of the leverage scores. For synthetic datasets, results agree with the observation made in (Ma et al., 2015) that leverage score sampling may outperform uniform sampling when dataset has highly skewed leverage scores. Order of the real datasets where leverage root does not perform well does not conform to this observation.

Randomized numerical linear algebra The methods from the field of randomized numerical linear algebra (Mahoney et al., 2011) has enabled solving numerical problems like linear least squares, low-rank approximation, and matrix multiplication for large data matrices. A central idea is to sketch the large matrices either by sampling or embedding them and perform computations on the resulting smaller matrices. A particular sampling-based approach named *leverage score sampling* (explained later) has yielded principled approximation methods. For the least squares problem in the offline setting where we observe A and b beforehand, sampling each row with probability proportional to their leverage scores provides a good approximation of the squared error (Drineas et al., 2006). These sampling methods provide $1+\epsilon$ approximation guarantee for the relative squared error with sample size $O(d \log d/\epsilon^2)$. In the online setting described above, Cohen et al. (2016) propose a variant of the leverage-score based sampling method with spectral approximation guarantees. For the more general problem of kernel regression, Calandriello et al. (2017) analyze approximation error for an online leverage score sampling method but they assume that only the design matrix is sub-sampled and notably *all* responses are sampled. Whereas in our setting responses have to be subsampled due to cost of acquiring them. In summary, we analyze leverage score sampling for the online least squares problem for the first time.

Active and online learning Our problem is an instance of the so-called stream-based active learning (e.g. (Dasgupta et al., 2009; Sabato and Hess, 2018)). However, the methods in this literature rely on properties of the data distribution and the model e.g. uniformly distributed data or linear separability (Dasgupta et al., 2009). Our analysis does not make such assumptions. As long as one is interested in the linear least squares solution, leverage score sampling method ensures good approximation error compared to that solution without relying on distributional assumptions. The related problem of online selective sampling (e.g. (Hanneke and Yang, 2021)) differs in the respect that one has to predict labels for each data point in the stream and algorithms are evaluated based on number of mistakes made and number of points sampled.

References

Les Atlas, David Cohn, and Richard Ladner. Training connectionist networks with queries and selective sampling. In D. Touretzky, editor, *Advances in Neural Information Processing Systems*, volume 2. Morgan-Kaufmann, 1989. URL `https://proceedings.neurips.cc/paper/1989/file/b1a59b315fc9a3002ce38bbe070ec3f5-Paper.pdf`.

Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2):235–256, 2002.

Katy S. Azoury and M. K. Warmuth. Relative loss bounds for on-line density estimation with the exponential family of distributions. In *MACHINE LEARNING*, page 2001. Morgan Kaufmann, 2000.

Daniele Calandriello, Alessandro Lazaric, and Michal Valko. Distributed Adaptive Sampling for Kernel Matrix Approximation. In Aarti Singh and Jerry Zhu, editors, *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pages 1421–1429, Fort Lauderdale, FL, USA, 20–22 Apr 2017. PMLR. URL http://proceedings.mlr.press/v54/calandriello17a.html.

Giovanni Cavallanti, Nicolò Cesa-bianchi, and Claudio Gentile. Linear classification and selective sampling under low noise conditions. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems*, volume 21. Curran Associates, Inc., 2008. URL https://proceedings.neurips.cc/paper/2008/file/0777d5c17d4066b82ab86dff8a46af6f-Paper.pdf.

Nicolò Cesa-Bianchi, Claudio Gentile, and Francesco Orabona. Robust bounds for classification via selective sampling. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, page 121–128, New York, NY, USA, 2009. Association for Computing Machinery. ISBN 9781605585161. doi: 10.1145/1553374.1553390. URL https://doi.org/10.1145/1553374.1553390.

Xue Chen and Eric Price. Active regression via linear-sample sparsification. In Alina Beygelzimer and Daniel Hsu, editors, *Proceedings of the Thirty-Second Conference on Learning Theory*, volume 99 of *Proceedings of Machine Learning Research*, pages 663–695. PMLR, 25–28 Jun 2019. URL https://proceedings.mlr.press/v99/chen19a.html.

Yining Chen, Haipeng Luo, Tengyu Ma, and Chicheng Zhang. Active online learning with hidden shifting domains. In Arindam Banerjee and Kenji Fukumizu, editors, *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pages 2053–2061. PMLR, 13–15 Apr 2021. URL https://proceedings.mlr.press/v130/chen21d.html.

Michael B Cohen, Yin Tat Lee, Cameron Musco, Christopher Musco, Richard Peng, and Aaron Sidford. Uniform sampling for matrix approximation. In *Proceedings of the 2015 Conference on Innovations in Theoretical Computer Science*, pages 181–190, 2015.

Michael B. Cohen, Cameron Musco, and Jakub Pachocki. Online row sampling. In *The 19th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems, APPROX 2016*, 2016.

Sanjoy Dasgupta, Adam Tauman Kalai, and Claire Monteleoni. Analysis of perceptron-based active learning. *Journal of Machine Learning Research*, 10(11):281–299, 2009. URL http://jmlr.org/papers/v10/dasgupta09a.html.

Ofer Dekel, Claudio Gentile, and Karthik Sridharan. Selective sampling and active learning from single and multiple teachers. *Journal of Machine Learning Research*, 13(86):2655–2697, 2012. URL http://jmlr.org/papers/v13/dekel12b.html.

Michal Derezinski, Manfred KK Warmuth, and Daniel J Hsu. Leveraged volume sampling for linear regression. In *Advances in Neural Information Processing Systems*, pages 2505–2514, 2018.

Petros Drineas and Michael W Mahoney. Lectures on randomized numerical linear algebra. *The Mathematics of Data*, 25:1, 2018.

Petros Drineas, Michael W Mahoney, and Shan Muthukrishnan. Sampling algorithms for l 2 regression and applications. In *Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm*, pages 1127–1136. Society for Industrial and Applied Mathematics, 2006.

Steve Hanneke and Liu Yang. Toward a general theory of online selective sampling: Trading off mistakes and queries. In Arindam Banerjee and Kenji Fukumizu, editors, *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pages 3997–4005. PMLR, 13–15 Apr 2021. URL `http://proceedings.mlr.press/v130/hanneke21a.html`.

Ping Ma, Michael W Mahoney, and Bin Yu. A statistical perspective on algorithmic leveraging. *The Journal of Machine Learning Research*, 16(1):861–911, 2015.

Ping Ma, Xinlian Zhang, Xin Xing, Jingyi Ma, and Michael Mahoney. Asymptotic analysis of sampling estimators for randomized numerical linear algebra algorithms. In Silvia Chiappa and Roberto Calandra, editors, *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 1026–1035. PMLR, 26–28 Aug 2020. URL `https://proceedings.mlr.press/v108/ma20b.html`.

Michael W Mahoney et al. Randomized algorithms for matrices and data. *Foundations and Trends® in Machine Learning*, 3(2):123–224, 2011.

Cem Orhan and Oznur Tastan. Active learning methods based on statistical leverage scores, 2018.

Carlos Riquelme, Ramesh Johari, and Baosen Zhang. Online active linear regression via thresholding. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, page 2506–2512. AAAI Press, 2017.

Sivan Sabato and Tom Hess. Interactive algorithms: Pool, stream and precognitive stream. *Journal of Machine Learning Research*, 18(229):1–39, 2018. URL `http://jmlr.org/papers/v18/16-424.html`.

Sivan Sabato and Remi Munos. Active regression by stratification. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014. URL `https://proceedings.neurips.cc/paper/2014/file/6883966fd8f918a4aa29be29d2c386fb-Paper.pdf`.

Volodya Vovk. Competitive on-line linear regression. In M. Jordan, M. Kearns, and S. Solla, editors, *Advances in Neural Information Processing Systems*, volume 10. MIT Press, 1997. URL `https://proceedings.neurips.cc/paper/1997/file/30c8e1ca872524fbf7ea5c519ca397ee-Paper.pdf`.

David P Woodruff. Sketching as a tool for numerical linear algebra. *arXiv preprint arXiv:1411.4357*, 2014.