Attribute Efficient Linear Regression with Distribution-Dependent Sampling

Doron Kukliansky DORON.KUKLIANSKY@WEIZMANN.AC.IL
Ohad Shamir OHAD.SHAMIR@WEIZMANN.AC.IL
Department of Computer Science and Applied Mathematics, Weizmann Institute of Science, Rehovot, Israel

Abstract

We consider a budgeted learning setting, where the learner can only choose and observe a small subset of the attributes of each training example. We develop efficient algorithms for Ridge and Lasso linear regression, which utilize the geometry of the data by a novel distribution-dependent sampling scheme, and have excess risk bounds which are better a factor of up to $O(\sqrt{d/k})$ over the state-of-the-art, where d is the dimension and $k+1$ is the number of observed attributes per example. Moreover, under reasonable assumptions, our algorithms are the first in our setting which can provably use *less* attributes than full-information algorithms, which is the main concern in budgeted learning. We complement our theoretical analysis with experiments which support our claims.

1. Introduction

Consider the problem of medical diagnosis, in which the learner wishes to determine whether a patient has some disease based on a series of medical tests. In order to build a model, the learner has to gather a set of volunteers, perform diagnostic tests on them and use the tests results as features. However, some of the volunteers may be reluctant to undergo a large number of tests, as medical tests may cause physical discomfort, and will prefer to undergo only a small number of them. During prediction time, however, patients are more likely to agree to undergo all tests, to find a diagnosis to their illness.

This problem is an example of budgeted learning (Madani et al., 2004) or learning with limited attribute observation (LAO) (Ben-David & Dichterman, 1993). Formally, we use the local budget setting presented in (Cesa-Bianchi et al., 2011): For each training example (composed of a d-dimensional attribute vector $\mathbf{x}$ and a target value y), we have a budget of $k+1$ attributes, where $k \ll d$, and we are able to choose which $k+1$ attributes we wish to reveal. Our goal is to find a good predictor that uses all the attributes despite the partial information at training time.

This problem has been previously studied in (Cesa-Bianchi et al., 2011; Hazan & Koren, 2012), in the context of linear predictors and the squared loss, under both L_2 (Ridge) and L_1 (Lasso) norm constraints (see also (Zolghadr et al., 2013) for a related but different setting, where the cost of observing attributes is incorporated into the loss function). Their algorithmic approach is based on online/stochastic gradient descent, using unbiased gradients estimates of the loss w.r.t. each example. The gradient estimator requires uniform sampling of attributes (up to the budget constraint), eventually leading to algorithms with expected excess risk bounds over the optimal predictor in the hypothesis class of $\widetilde{\mathrm{O}}\left(\sqrt{(d/k)/m}\right)$ after m examples, compared with $\widetilde{\mathrm{O}}\left(\sqrt{1/m}\right)$ for full-information algorithms that can view all the attributes (Kakade et al., 2009) (see Table 1). Another interpretation of these results is that even though the algorithms view only $k+1$ out of d attributes, the algorithms need the same total number of attributes, $\widetilde{\mathrm{O}}\left(d/\epsilon^2\right)$, to obtain the same accuracy ϵ. Moreover, (Cesa-Bianchi et al., 2011; Hazan & Koren, 2012) provide a lower bound establishing that Ridge bound is not improvable in general.

In this paper, despite these seemingly unimprovable results, we show that they can in fact be improved. We do this by developing a novel sampling scheme which samples the attributes in a *distribution-dependent* manner: We sample attributes with large second moments more than others, thus gaining a distribution-dependent improvement factor. In other words, our sampling methods take advantage of the geometry of the data distribution, and utilize it to extract more 'information' out of each sample. Under reasonable assumptions, our methods need *less* attributes to reach the same accuracy than the online full-information algorithms, which is beneficial in budgeted learning scenarios. As far as we know, these are the first methods provably able to do so in our setting.

Proceedings of the 32^{nd} International Conference on Machine Learning, Lille, France, 2015. JMLR: W&CP volume 37.

We begin by assuming prior knowledge of the second moments of the attribute vector, namely $\mathbb{E}_D\left[x_i^2\right]$ for $i \in [d]$, where we use $\mathbb{E}_D[\cdot]$ to denote the expectation with respect to the data distribution. Our excess risk bounds are summarized in Table 1. To clarify the notation, $\left\|\mathbb{E}_D\left[\mathbf{x}^2\right]\right\|_{\frac{1}{2}}$ is defined as $\left(\sum_{i=1}^d \sqrt{\mathbb{E}_D\left[x_i^2\right]}\right)^2$, and $\left\|\mathbb{E}_D\left[\mathbf{x}^2\right]\right\|_1$ is defined as $\sum_{i=1}^d \mathbb{E}_D\left[x_i^2\right]$.

Table 1. Expected excess risk bounds assuming $\|\mathbf{x}\|_2 \leq 1$ in the Ridge scenario and $\|\mathbf{x}\|_\infty \leq 1$ in the Lasso scenario.

Type	New Bound	Old Bound	Full-Information Online Bound
Ridge	$\mathrm{O}\left(\sqrt{\frac{\left\|\mathbb{E}_D[\mathbf{x}^2]\right\|_{\frac{1}{2}}+k}{km}}\right)$	$\mathrm{O}\left(\sqrt{\frac{d}{km}}\right)$	$\mathrm{O}\left(\sqrt{\frac{1}{m}}\right)$
Lasso	$\mathrm{O}\left(\sqrt{\frac{\left(\left\|\mathbb{E}_D[\mathbf{x}^2]\right\|_1+k\right)\log d}{km}}\right)$	$\mathrm{O}\left(\sqrt{\frac{d\log d}{km}}\right)$	$\mathrm{O}\left(\sqrt{\frac{\log d}{m}}\right)$

It can be easily shown that under the relevant data norm constraints, both $\left\|\mathbb{E}_D\left[\mathbf{x}^2\right]\right\|_{\frac{1}{2}}$ and $\left\|\mathbb{E}_D\left[\mathbf{x}^2\right]\right\|_1$ are at most d, which proves that our bounds are always as good as the previous bounds. In fact, the equalities hold only when all second moments are equal. Otherwise, both values are strictly smaller than d, making our bounds better. This improvement factor is distribution-dependent and may be as large as $\mathrm{O}\left(\sqrt{d/k}\right)$ (i.e. both values can be $\mathrm{O}(1)$) when the second moments decay sufficiently fast. We note that similar distributional assumptions about moment decay are made in other successful algorithmic approaches such as AdaGrad (Duchi et al., 2011). When the attribute budget satisfies $k = \Omega\left(\left\|\mathbb{E}_D\left[\mathbf{x}^2\right]\right\|_{\frac{1}{2}}\right)$ (or $k = \Omega\left(\left\|\mathbb{E}_D\left[\mathbf{x}^2\right]\right\|_1\right)$ in the Lasso scenario) our bounds also coincide with the online full-information scenario.

When no such prior knowledge is available, we split our algorithms into two phases: In the first phase, we estimate a certain upper bound on the second moments of the attributes. In the second phase, we use the same sampling scheme but with smoothed probabilities, to compensate for the stochastic error in the estimation phase. We prove that the excess risk bound of this method is always as good as those of (Hazan & Koren, 2012), and given sufficient training examples, achieves the same bounds as our algorithms which assume prior knowledge of the moments (up to constant factors).

2. Preliminaries

2.1. Notation

We indicate scalars by a small letter, a, and vectors by a bold font, $\mathbf{a}$. We use $\mathbf{a}^2$ to indicate the vector for which $\mathbf{a}^2[i] = a[i]^2$, and $\mathbf{a} + b$ to indicate the vector for which $(\mathbf{a}+b)[i] = a[i] + b$. We denote the i-th vector of the standard basis by $\mathbf{e}_i$. We indicate the set of indices $1, .., n$ by $[n]$. We use $\|\mathbf{a}\|_p$ to indicate the p-norm of the vector, $\left(\sum_{i=1}^d |a_i|^p\right)^{\frac{1}{p}}$. We apply this notation also for the case where $p = \frac{1}{2}$ i.e. $\|\mathbf{a}\|_{\frac{1}{2}} = (\sum_{i=1}^d \sqrt{|a_i|})^2$, even though this is not a proper norm. We also use $\|\mathbf{a}\|_\infty$ to indicate the infinity norm, $\max_i |a_i|$. We denote the expectation with respect to the randomness of the algorithm (attribute sampling) by $\mathbb{E}_A[\cdot]$, the expectation with respect to the data distribution by $\mathbb{E}_D[\cdot]$ and the expectation with respect to both by $\mathbb{E}_{D,A}[\cdot]$. For the two-phased algorithms, we use $\mathbb{E}_{D,A_i}[\cdot]$ where $i \in \{1,2\}$ to denote the expectation with respect to the data distributions and the randomness of the algorithm during the i-th phase. We denote the loss induced by the t-th example in the training set as $\ell_t(\mathbf{w})$.

2.2. Linear Regression

Following the standard framework for statistical learning, we assume the training set, $\left\{(\mathbf{x}_t, y_t) \in \mathbb{R}^d \times \mathbb{R}\right\}_{t=1}^m$, was sampled i.i.d. from some joint distribution $\mathcal{D}$. Each $\mathbf{x}_t$ is a data point, represented by a vector of attributes, and y_t is the desired target value. The goal of the learner is to find a weight vector $\mathbf{w}$, such that $\hat{y}_t = \langle \mathbf{w}, \mathbf{x}_t \rangle$ is a good estimator of y_t, in the sense that it minimizes the expected loss, or the risk, $L_{\mathcal{D}}(\mathbf{w}) = \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\left[\ell\left(\mathbf{w}^T\mathbf{x}, y\right)\right]$. Here we focus on the squared loss i.e. $\ell(\hat{y}, y) = \frac{1}{2}(\hat{y} - y)^2$.

To prevent overfitting, it is common practice to constrain the norm of $\mathbf{w}$. We designate the 2-norm case, where we want to find a good predictor in $\mathcal{F} = \{\mathbf{w} | \|\mathbf{w}\|_2 \leq B\}$, as the Ridge regression scenario, and the 1-norm case, where we consider $\{\mathbf{w} | \|\mathbf{w}\|_1 \leq B\}$, as the Lasso regression scenario. We will assume w.l.o.g. that $\|\mathbf{x}\|_2 \leq 1$ in the Ridge scenario, and $\|\mathbf{x}\|_\infty \leq 1$ in the Lasso scenario, and that $|y_t| \leq B$ in both cases.

In the full-information scenario, the learner has access to all the attributes of $\mathbf{x}_t$, whereas in the attribute efficient scenario, the learner can choose $k+1$ attributes out of d from each vector $\mathbf{x}_t$ in the training set.

3. Attribute Efficient Ridge Regression

In this section we present our algorithms for Ridge regression where the 2-norm is bounded, $\|\mathbf{w}\|_2 \leq B$. The generic approach to the Ridge attribute efficient scenario, which we call the General Attribute Efficient Ridge Regression (GAERR) algorithm and is presented in Algorithm 1, was developed in (Cesa-Bianchi et al., 2011; Hazan & Koren, 2012) and is based on the Online Gradient Descent (OGD) algorithm with gradient estimates.

The OGD algorithm goes over the training set, and for each example builds an unbiased estimator of the gradient. Af-

terwards, the algorithm updates the current weight vector, $\mathbf{w}_t$, by performing a step of size η in the opposite direction to the gradient estimator. The result is projected over the L_2 ball of size B, yielding $\mathbf{w}_{t+1}$. At the end, the algorithm outputs the average of all $\mathbf{w}_t$.

The gradient of the squared loss is $\nabla \ell(\mathbf{w}; \mathbf{x}_t, y_t) = (\langle \mathbf{w}, \mathbf{x}_t \rangle - y_t) \cdot \mathbf{x}_t$, and the key idea of the GAERR algorithm is how to use the budgeted sampling to construct an unbiased estimator for the gradient. It does so by sampling $k+1$ attributes out of the d attributes of the sample where $k > 0$ is the a budget parameter[1]: First, it samples k attributes with probabilities q_i with replacement, and by weighting them correctly, builds an unbiased estimator for the data point, $\widetilde{\mathbf{x}}_t$. Afterwards, it samples an additional attribute with probability $p_{j_t} = w_{t,j_t}^2 / \|\mathbf{w}_t\|_2^2$ and by a simple calculation obtains an unbiased estimator of the inner product. Subtracting the label, y_t, yields the unbiased estimator, $\widetilde{\phi}_t$. Finally, the algorithms multiplies the two parts, thus building an unbiased estimator of the gradient for the point, $\widetilde{\mathbf{g}}_t$.

Algorithm 1 GAERR

Parameters: $B, \eta > 0$ and q_i for $i \in [d]$

Input: training set $S = \{(\mathbf{x}_t, y_t)\}_{t \in [m]}$ and $k > 0$

Output: regressor $\bar{\mathbf{w}}$ with $\|\bar{\mathbf{w}}\|_2 \leq B$

Initialize $\mathbf{w}_1 \neq 0$, $\|\mathbf{w}_1\|_2 \leq B$ arbitrarily
for $t = 1$ to m **do**
  for $r = 1$ to k **do**
    Pick $i_{t,r} \in [d]$ with probability $q_{i_{t,r}}$ and observe $\mathbf{x}_t[i_{t,r}]$
    $\widetilde{\mathbf{x}}_{t,r} \leftarrow \frac{1}{q_{i_{t,r}}} \mathbf{x}_t[i_{t,r}] \mathbf{e}_{i_{t,r}}$
  end for
  $\widetilde{\mathbf{x}}_t \leftarrow \frac{1}{k} \sum_{r=1}^{k} \widetilde{\mathbf{x}}_{t,r}$
  Choose $j_t \in [d]$ with probability $p_{j_t} = \frac{w_{t,j_t}^2}{\|\mathbf{w}_t\|_2^2}$ and observe $\mathbf{x}_t[j_t]$
  $\widetilde{\phi}_t \leftarrow \frac{w_{t,j}}{p_{j_t}} \mathbf{x}_t[j_t] - y_t$
  $\widetilde{\mathbf{g}}_t \leftarrow \widetilde{\phi}_t \cdot \widetilde{\mathbf{x}}_t$
  $\mathbf{v}_t \leftarrow \mathbf{w}_t - \eta \widetilde{\mathbf{g}}_t$
  $\mathbf{w}_{t+1} \leftarrow \mathbf{v}_t \cdot \frac{B}{\max\{\|\mathbf{v}_t\|_2, B\}}$
end for
$\bar{\mathbf{w}} \leftarrow \frac{1}{m} \sum_{t=1}^{m} \mathbf{w}_t$

The expected excess risk bound of the GAERR algorithm is presented in the next theorem which is a slightly more general version of Theorem 3.1 in (Hazan & Koren, 2012).

Theorem 3.1. *Assume the distribution $\mathcal{D}$ is such that $\|\mathbf{x}\|_2 \leq 1$ and $|y| \leq B$ with probability 1. Let $\bar{\mathbf{w}}$ be the output of GAERR when run with step size η and let $\max_t \mathbb{E}_{D,A}\left[\|\widetilde{\mathbf{g}}_t\|_2^2\right] \leq G^2$. Then for any $\mathbf{w}^* \in \mathbb{R}^d$ with $\|\mathbf{w}^*\|_2 \leq B$,*

$$\mathbb{E}_{D,A}\left[L_{\mathcal{D}}(\bar{\mathbf{w}})\right] \leq L_{\mathcal{D}}(\mathbf{w}^*) + \frac{2B^2}{\eta m} + \frac{\eta}{2} G^2.$$

The intuition is that it OGD requires merely unbiased gradient estimates, as long as their second moments, G, are bounded. The full proof can be found in Appendix C.1.

The AERR algorithm is one variant of the GAERR algorithm. It was presented in (Hazan & Koren, 2012) and uses uniform sampling to estimate $\mathbf{x}_t$. In our GAERR notation it uses $q_i = \frac{1}{d} \quad \forall i \in [d]$. The authors prove (Lemma 3.3 in (Hazan & Koren, 2012)) that for the AERR algorithm, $G^2 \leq 8B^2 d/k$, which together with Theorem 3.1 and using $\eta = 2B/G\sqrt{m}$ yields an expected excess risk bound of $4B^2\sqrt{2d/km}$. They also prove that their algorithm is optimal up to constant factors (in the worst-case over all data distributions), by showing a corresponding lower bound.

This, however, is not the end of the story. By analyzing the bound, we show that we can improve it in a distribution-dependent manner. Theorem 3.1 shows us that the expected excess risk bound is proportional to G, therefore we wish to develop a sampling method that allows us to minimize $\mathbb{E}_{D,A}\left[\|\widetilde{\mathbf{g}}_t\|_2^2\right]$, as stated in the next lemma.

Lemma 3.2. *The GAERR algorithm generates gradient estimates that for all t, $\mathbb{E}_{D,A}\left[\|\widetilde{\mathbf{g}}_t\|_2^2\right] \leq 4B^2\left(\frac{1}{k}\mathbb{E}_{D,A}\left[\|\widetilde{\mathbf{x}}_{t,r}\|_2^2\right] + 1\right)$.*

The proof can be found in Appendix C.2.

Since

$$\mathbb{E}_{D,A}\left[\|\widetilde{\mathbf{x}}_{t,r}\|_2^2\right] = \mathbb{E}_{D,A}\left[\widetilde{\mathbf{x}}_{t,r}[i_{t,r}]^2\right] = \sum_{i=1}^{d} \frac{1}{q_i} \mathbb{E}_D\left[x_i^2\right], \tag{1}$$

we can minimize this bound as a function of the q_i-s, under the constraints of $\sum_{i=1}^{d} q_i = 1$ and $q_i \geq 0$ for all $i \in [d]$. This optimization problem can easily be solved using Lagrange multipliers to yield the solution

$$q_i = \frac{\sqrt{\mathbb{E}_D\left[x_i^2\right]}}{\sum_{j=1}^{d} \sqrt{\mathbb{E}_D\left[x_j^2\right]}}. \tag{2}$$

We could have followed a similar optimization strategy for finding the optimal sampling distribution for estimating the inner product. This strategy would have yielded that the optimal probabilities are $p_{j_t} = \frac{\sqrt{\mathbf{w}_{t,j_t}^2 \mathbb{E}_D\left[x_{j_t}^2\right]}}{\sum_{l=1}^{d} \sqrt{\mathbf{w}_{t,l}^2 \mathbb{E}_D\left[x_l^2\right]}}$. However, this does not materially improve the analysis, and is therefore not included.

[1] As in the AERR algorithm, we assume we have a budget of at least 2 attributes per training sample.

3.1. Known Second Moment Scenario

If we assume prior knowledge of the second moment of each attribute, namely $\mathbb{E}_D\left[x_i^2\right]$ for all $i \in [d]$, we can use equation (2) to calculate the optimal values of the q_i-s. This is the idea behind our DDAERR (Distribution-Dependent Attribute Efficient Ridge Regression) algorithm. Its expected excess risk bound is formulated in the next theorem.

Theorem 3.3. *Assume the distribution $\mathcal{D}$ is such that $\|\mathbf{x}\|_2 \leq 1$ and $|y| \leq B$ with probability 1[2] and $\mathbb{E}_D\left[x_i^2\right]$ are known for $i \in [d]$. Let $\bar{\mathbf{w}}$ be the output of DDAERR, when run with $\eta = \frac{1}{\sqrt{m\left(\frac{1}{k}\|\mathbb{E}_D[\mathbf{x}^2]\|_{\frac{1}{2}}+1\right)}}$. Then for any $\mathbf{w}^* \in \mathbb{R}^d$ with $\|\mathbf{w}^*\|_2 \leq B$,*

$$\mathbb{E}_{D,A}\left[L_{\mathcal{D}}\left(\bar{\mathbf{w}}\right)\right] \leq L_{\mathcal{D}}\left(\mathbf{w}^*\right) + 4\frac{B^2}{\sqrt{m}}\sqrt{\frac{1}{k}\left\|\mathbb{E}_D\left[\mathbf{x}^2\right]\right\|_{\frac{1}{2}}+1}.$$

The proof can be found in Appendix C.3.

Recalling that with probability 1 we have $\|\mathbf{x}\|_2 \leq 1$, it is easy to see that $\left\|\mathbb{E}_D\left[\mathbf{x}^2\right]\right\|_{\frac{1}{2}} \leq d$, therefore the DDAERR bound is at least as good as the AERR bound[3]. However, $\left\|\mathbb{E}_D\left[\mathbf{x}^2\right]\right\|_{\frac{1}{2}}$ may also be much smaller than d, in cases where the second moments vary between attributes or the $\mathbb{E}_D\left[\mathbf{x}^2\right]$ is approximately sparse. In these cases, we may gain a significant improvement.

3.2. Unknown Second Moment Scenario

The solution presented in the previous section requires exact knowledge of $\mathbb{E}_D\left[x_i^2\right]$ for all i. Such prior knowledge may not be available, thus we turn to consider the case where the moments are initially unknown. The problem in this scenario is that without prior knowledge of the second moments of the attributes, the learner cannot calculate the optimal q_i-s via equation (2). To address this issue we split the learning into two phases: In the first phase we run on the first m_1 training examples and estimate the second moments by sampling the attributes uniformly at random. In the second phase we run on the next m_2 training examples, and perform the regular DDAERR algorithm, with a slight modification - in the calculation of the q_i-s, we use an upper confidence interval instead of the second moments themselves, namely $q_i = \frac{\sqrt{\mathbf{A}[i]+\frac{13}{6}\epsilon}}{\sum_{j=1}^{d}\sqrt{\mathbf{A}[j]+\frac{13}{6}\epsilon}}$ where $A[i]$ is the average of the square of the i-th attribute as calculated during the first phase, $\epsilon = \frac{d\log(2d/\delta)}{(k+1)m_1}$ and δ is the probability parameter. This approach is the basis for our Two-Phased DDAERR algorithm (Algorithm 2 in Appendix A.1).

In practice, one can run the AERR algorithm during the first phase, in order to obtain a better starting point for the second phase. However, We ignore this improvement in our analysis, but incorporate it in the experiments presented in section 5.

The expected excess risk bound of the algorithm is formulated in the following theorem.

Theorem 3.4. *Assume the distribution $\mathcal{D}$ is such that $\|\mathbf{x}\|_2 \leq 1$ and $|y| \leq B$ with probability 1. Let $\bar{\mathbf{w}}$ be the output of Two-Phased DDAERR when run with $\eta = \max(\eta_1, \eta_2)$ where $\eta_1 = \sqrt{\frac{k}{6dm_2}}$ and $\eta_2 = \sqrt{\frac{k/m_2}{2\left\|2\mathbf{A}+\frac{10}{3}\epsilon\right\|_{\frac{1}{2}}+2\sqrt{\frac{5d^3\left\|2\mathbf{A}+\frac{10}{3}\epsilon\right\|_{\frac{1}{2}}\log\frac{2d}{\delta}}{3(k+1)m_1}}+k}}$. Then for all m_1 and for any $\mathbf{w}^* \in \mathbb{R}^d$ with $\|\mathbf{w}^*\|_2 \leq B$, with probability 1 over the first phase, we have*

$$\mathbb{E}_{D,A_2}\left[L_{\mathcal{D}}\left(\bar{\mathbf{w}}\right)\right] - L_{\mathcal{D}}\left(\mathbf{w}^*\right) \leq \frac{4B^2}{\sqrt{m_2}}\sqrt{\frac{6d}{k}}.$$

Also, with probability $\geq 1-\delta$ over the first phase, we have

$$\mathbb{E}_{D,A_2}\left[L_{\mathcal{D}}\left(\bar{\mathbf{w}}\right)\right] - L_{\mathcal{D}}\left(\mathbf{w}^*\right) \leq \frac{16B^2}{\sqrt{m_2}}\sqrt{\frac{1}{k}\left(\sqrt{\left\|\mathbb{E}_D\left[\mathbf{x}^2\right]\right\|_{\frac{1}{2}}} + d\sqrt{\frac{2d\log\frac{2d}{\delta}}{(k+1)m_1}}\right)^2 + 1}.$$

The proof can be found in Appendix C.4.

If we examine the bound we can see that with probability 1 over the first phase, regardless of the value of m_1, the expected excess risk bound is at most $\mathrm{O}\left(\frac{B^2}{\sqrt{km_2}}\sqrt{d}\right)$, which is the same bound as the AERR algorithm. As m_1 increases, the bound turns to $\mathrm{O}\left(\frac{B^2}{\sqrt{km_2}}\sqrt{\left(\sqrt{\left\|\mathbb{E}_D\left[\mathbf{x}^2\right]\right\|_{\frac{1}{2}}} + d\sqrt{\frac{d\log\frac{2d}{\delta}}{(k+1)m_1}}\right)^2 + k}\right)$. Therefore, if $m_1 \gg \frac{d^2\log\frac{2d}{\delta}}{k+1}$, we achieve an improvement over the AERR algorithm. If $m_1 \geq \frac{d^3\log\frac{2d}{\delta}}{(k+1)\|\mathbb{E}_D[\mathbf{x}^2]\|_{\frac{1}{2}}}$, the bound becomes $\mathrm{O}\left(\frac{B^2}{\sqrt{km_2}}\sqrt{\left\|\mathbb{E}_D\left[\mathbf{x}^2\right]\right\|_{\frac{1}{2}}+k}\right)$, which is the same bound as in the regular DDAERR algorithm with prior knowledge of the second moment of the attributes.

4. Attribute Efficient Lasso Regression

In this section we present our algorithms for Lasso regression, where the loss is again the squared loss, but this time the 1-norm is bounded, i.e. $\|\mathbf{w}\|_1 \leq B$.

[2] Actually, in all the relevant locations, it is enough to assume only $\mathbb{E}_D\left[y^2\right]$ is bounded, but we prefer to remain within the framework of previous works.

[3] If $\left\|\mathbb{E}_D\left[\mathbf{x}^2\right]\right\|_{\frac{1}{2}} = d$ we have that $\mathbb{E}_D\left[x_i\right] = \frac{1}{d}$ for all $i \in [d]$. In this case, all the q_i-s are equal to $\frac{1}{d}$ and the DDAERR and AERR algorithms coincide.

The generic approach to the Lasso attribute efficient scenario, which we call the General Attribute Efficient Lasso Regression (GAELR) algorithm, is similar to the Ridge scenario but with two main differences: First, it is based on a variant of the Exponentiated Gradient (EG) algorithm using unbiased gradient estimates (Kivinen & Warmuth, 1997; Hazan & Koren, 2012), instead of the OGD algorithm. Second, when estimating the inner product, instead of sampling one attribute with probability $p_{j_t} = w_{t,j_t}^2 / \|\mathbf{w}_t\|_2^2$, it samples it with probability $p_{j_t} = |w_{t,j_t}| / \|\mathbf{w}_t\|_1$, as the Lasso scenario has a bound on the 1-norm of the predictor. The rest of the estimation process is the same. More details can be found in Appendix A.2.

The expected excess risk bound of the GAELR algorithm is presented in the next theorem which is a slightly more general version of Theorem 3.4 in (Hazan & Koren, 2012).

Theorem 4.1. *Assume the distribution $\mathcal{D}$ is such that $\|\mathbf{x}\|_\infty \leq 1$ and $|y| \leq B$ with probability 1. Let $\bar{\mathbf{w}}$ be the output of GAELR, when run with step size $\eta \leq \frac{1}{2G}$ where $\max_t \left\|\mathbb{E}_{D,A}\left[\widetilde{\mathbf{g}}_t^{\,2}\right]\right\|_\infty \leq G^2$. Then for any $\mathbf{w}^* \in \mathbb{R}^d$ with $\|\mathbf{w}^*\|_1 \leq B$,*

$$\mathbb{E}_{D,A}\left[L_{\mathcal{D}}\left(\bar{\mathbf{w}}\right)\right] \leq L_{\mathcal{D}}\left(\mathbf{w}^*\right) + B\left(\frac{\log 2d}{\eta m} + 5\eta G^2\right).$$

The general idea of the proof is that $\widetilde{\mathbf{g}}_t$ is an unbiased estimator of the gradient, therefore we can use the standard analysis of the EG algorithm. The full proof can be found in Appendix C.10.

The AELR algorithm is one variant of the GAELR algorithm. It was presented in (Hazan & Koren, 2012) and uses uniform sampling to estimate $\mathbf{x}_t$. In our GAELR notation it uses $q_i = \frac{1}{d} \quad \forall i \in [d]$. The authors prove (Lemma 3.8 in (Hazan & Koren, 2012)) that for the AELR algorithm, $G^2 \leq 8B^2 d/k$, which together with Theorem 4.1 and using $\eta = \frac{2B}{G\sqrt{m}}$ yields an expected excess risk bound of $4B^2\sqrt{\frac{10d\log 2d}{km}}$.

Similarly to the Ridge scenario, by analyzing the bound, we show that we can improve the bound in a distribution-dependent manner: Theorem 4.1 tells us that the expected excess risk bound is proportional to G, therefore we wish to develop a sampling method that minimizes the infinity norm of the gradient estimator.

Lemma 4.2. *The GAELR algorithm generates gradient estimates that for all t, $\left\|\mathbb{E}_{D,A}\left[\widetilde{\mathbf{g}}_t^{\,2}\right]\right\|_\infty \leq 4B^2\left(\frac{1}{k}\left\|\mathbb{E}_{D,A}\left[\widetilde{\mathbf{x}}_{t,r}^2\right]\right\|_\infty + 1\right)$.*

The proof can be found in Appendix C.11.

Since

$$\mathbb{E}_{D,A}\left[\widetilde{\mathbf{x}}_{t,r}^2[i]\right] = \frac{1}{q_i}\mathbb{E}_D\left[x_i^2\right], \tag{3}$$

we can minimize this bound as a function of the q_i-s, under the constraints of $\sum_{i=1}^d q_i = 1$ and $q_i \geq 0$ for all $i \in [d]$.

Lemma 4.3. *The solution to the optimization problem defined is $q_i = \frac{\mathbb{E}_D\left[x_i^2\right]}{\sum_{j=1}^d \mathbb{E}_D\left[x_j^2\right]}$.*

The proof can be found in Appendix C.12.

As in the Ridge scenario, we could have tried to optimize the sampling probabilities of the inner product estimation. However, since $\mathbb{E}_{D,A}\left[\widetilde{\phi}_t^{\,2}\right]$ is calculated using the same method as in the Ridge scenario, the optimal sampling probabilities remain $p_{j_t} = \frac{\sqrt{\mathbf{w}_{t,j_t}^2 \mathbb{E}_D\left[x_{j_t}^2\right]}}{\sum_{l=1}^d \sqrt{\mathbf{w}_{t,l}^2 \mathbb{E}_D\left[x_l^2\right]}}$, but we will still not include this improvement in our analysis.

4.1. Known Second Moment Scenario

If we assume we have prior knowledge of the second moment of each attribute, we can use Lemma 4.3 to calculate the optimal values of the q_i-s. This is the idea behind our DDAELR (Distribution-Dependent Attribute Efficient Lasso Regression) algorithm. Its expected excess risk bound is formulated in the next theorem.

Theorem 4.4. *Assume the distribution $\mathcal{D}$ is such that $\|\mathbf{x}\|_\infty \leq 1$ and $|y| \leq B$ with probability 1 and $\mathbb{E}_D\left[x_i^2\right]$ are known for $i \in [d]$. Let $\bar{\mathbf{w}}$ be the output of DDAELR, when run with $\eta = \frac{1}{2B}\sqrt{\frac{\log 2d}{5m\left(\frac{1}{k}\|\mathbb{E}_D[\mathbf{x}^2]\|_1 + 1\right)}}$. If $m \geq \log 2d$ then for any $\mathbf{w}^* \in \mathbb{R}^d$ with $\|\mathbf{w}^*\|_1 \leq B$,*

$$\mathbb{E}_{D,A}\left[L_{\mathcal{D}}\left(\bar{\mathbf{w}}\right)\right] - L_{\mathcal{D}}\left(\mathbf{w}^*\right) \leq 4B^2\sqrt{\frac{5\log 2d\left(\|\mathbb{E}_D\left[\mathbf{x}^2\right]\|_1 + k\right)}{km}}.$$

The proof can be found in Appendix C.13.

Recalling that with probability 1 we have $\|\mathbf{x}\|_\infty \leq 1$, it is easy to see that $\left\|\mathbb{E}_D\left[\mathbf{x}^2\right]\right\|_1 \leq d$, therefore the DDAELR bound is at least as good as the AELR bound[4]. However, $\left\|\mathbb{E}_D\left[\mathbf{x}^2\right]\right\|_1$ may also be much smaller than d, in cases where the second moments vary between attributes or the vector is sparse. In these cases, we may gain a significant improvement.

4.2. Unknown Second Moment Scenario

In a case we lack prior knowledge of $\mathbb{E}_D\left[x_i^2\right]$ for all i, we take a similar approach to the Two-Phased DDAERR algorithm: in the first phase, we estimate the second moments by uniform sampling, exactly as in the Two-Phased DDAERR algorithm. In the second phase, we

[4] If $\left\|\mathbb{E}_D\left[\mathbf{x}^2\right]\right\|_1 = d$ we have that $\mathbb{E}_D\left[x_i\right] = 1$ for all $i \in [d]$. In this case, all the q_i-s are equal to $\frac{1}{d}$ and the DDAELR and AELR algorithms coincide.

run the DAELR with modified q_i-s, but this time with $q_i = \frac{\mathbf{A}[i]+\frac{13}{6}\epsilon}{\sum_{j=1}^{d}\left(\mathbf{A}[j]+\frac{13}{6}\epsilon\right)}$ which are more suitable for the Lasso scenario. This approach is the basis for our Two-Phased DDAELR algorithm (Algorithm 4 in Appendix A.3).

As in the Two-Phased DDAERR algorithm, during the first phase one can actually run the AELR algorithm in order to obtain a better starting point for the second phase, but again we will ignore this improvement in our analysis.

The expected excess risk bound of the algorithm is formulated in the following theorem.

Theorem 4.5. *Assume the distribution $\mathcal{D}$ is such that $\|\mathbf{x}\|_\infty \leq 1$ and $|y| \leq B$ with probability 1. Let $\bar{\mathbf{w}}$ be the output of DDAELR, when run with $\eta = \sqrt{\frac{k\log 2d}{20B^2 m_2\left(8\|\mathbf{A}\|_1+20d\min\left(\frac{d\log\frac{2d}{\delta}}{(k+1)m_1},1\right)+k\right)}}$. If $m_2 \geq \log 2d$ then for any m_1 and for any $\mathbf{w}^* \in \mathbb{R}^d$ with $\|\mathbf{w}^*\|_1 \leq B$, with probability 1 over the first phase we have*

$$\mathbb{E}_{D,A_2}\left[L_{\mathcal{D}}\left(\bar{\mathbf{w}}\right)\right] - L_{\mathcal{D}}\left(\mathbf{w}^*\right) \leq 61B^2\sqrt{\frac{d\log 2d}{km_2}}.$$

Also, with probability $1-\delta$ over the first phase we have

$$\mathbb{E}_{D,A_2}\left[L_{\mathcal{D}}\left(\bar{\mathbf{w}}\right)\right] - L_{\mathcal{D}}\left(\mathbf{w}^*\right) \leq 4B^2 \times \sqrt{\frac{5\left(16\left\|\mathbb{E}_D\left[\mathbf{x}^2\right]\right\|_1 + \frac{88d}{3}\min(\frac{d\log\frac{2d}{\delta}}{(k+1)m_1},1)+k\right)\log 2d}{km_2}}.$$

The proof can be found in Appendix C.14.

With probability 1 over the first phase, regardless of the value of m_1, the expected excess risk bound is at most $\mathrm{O}\left(\frac{B^2}{\sqrt{km_2}}\sqrt{d\log d}\right)$, which is the same bound of the AELR algorithm. As m_1 increases, the expected excess risk bound becomes $\mathrm{O}\left(\frac{B^2}{\sqrt{km_2}}\sqrt{\left(\left\|\mathbb{E}_D\left[\mathbf{x}^2\right]\right\|_1 + \frac{d^2\log\frac{2d}{\delta}}{(k+1)m_1}+k\right)\log d}\right)$. Therefore, if $m_1 \gg \frac{d\log\frac{2d}{\delta}}{k+1}$, we achieve an improvement over the AELR algorithm. If $m_1 \geq \frac{d^2\log\frac{2d}{\delta}}{(k+1)\left\|\mathbb{E}_D\left[\mathbf{x}^2\right]\right\|_1}$, the expected excess risk bound turns to $\mathrm{O}\left(\frac{B^2}{\sqrt{km_2}}\sqrt{\left\|\mathbb{E}_D\left[\mathbf{x}^2\right]\right\|_1+k}\right)$, which is the same bound as in the regular DDAELR algorithm with prior knowledge of the second moment of the attributes.

Interestingly, here the first phase generally requires less samples than the two-phased DDAERR algorithm. This is essentially due to $\mathbb{E}_D\left[x_i^2\right]$ being easier to estimate than $\sqrt{\mathbb{E}_D\left[x_i^2\right]}$, because the square root is not a Lipschitz function.

5. Experiments

We now turn to describe some experiments illustrating the behavior of our algorithms. We conducted two sets of experiments: One on artificial data, which allows us to easily control data properties such as $\left\|\mathbb{E}_D\left[\mathbf{x}^2\right]\right\|_{\frac{1}{2}}$ and $\left\|\mathbb{E}_D\left[\mathbf{x}^2\right]\right\|_1$; And the other on a subset of the popular MNIST (LeCun et al., 1998) data set, similar to (Cesa-Bianchi et al., 2011; Hazan & Koren, 2012). An additional experiment on a different data set is described in Appendix B.

In the Ridge regression scenario we tested 5 algorithms:

1. Our DDAERR algorithm that has prior knowledge of the second moment of the attributes.
2. Our Two-Phased DDAERR algorithm that does not have prior knowledge of the second moments of the attributes, and tries to estimate them.
3. The AERR algorithm that does not require any prior knowledge.
4. Online Ridge regression that performs online gradient descent and has access to all the attributes.
5. Offline Ridge regression that minimizes the empirical risk, which also has access to all attributes, and uses each training example more than once.

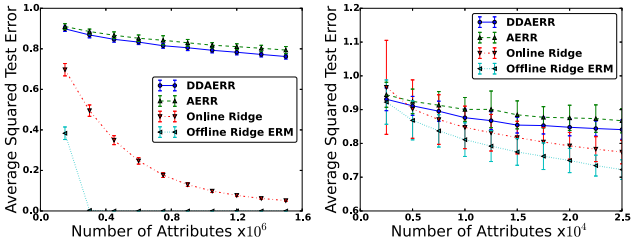
For the Lasso scenario we used the corresponding algorithms. In all cases our algorithms used the improved inner product estimation as well as the improved data point estimation.

For a fair comparison between the attribute efficient algorithms and the full-information algorithms, we use the X-axis in our figures to represent the number of *attributes* each algorithm sees, rather than the number of examples, since the comparison should be in terms of the total attribute budget used.

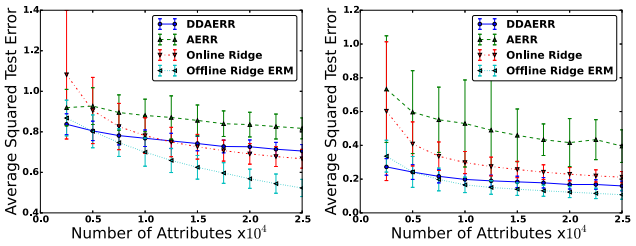
To quantify the theoretical improvement of the DDAERR algorithm, we compare $\left\|\mathbb{E}_D\left[\mathbf{x}^2\right]\right\|_{\frac{1}{2}}$ and $\left\|\mathbb{E}_D\left[\mathbf{x}^2\right]\right\|_1$ to d, as this is the potential improvement according to our analysis. To avoid scaling issues, we normalize by the 2-norm or the ∞-norm of the data and define our 'Improvement Ratios' by

$$\rho_{\text{Ridge}} = \frac{\left\|\mathbb{E}_D\left[\mathbf{x}^2\right]\right\|_{\frac{1}{2}}}{d\mathbb{E}_D\left[\|\mathbf{x}\|_2^2\right]}, \quad \rho_{\text{Lasso}} = \frac{\left\|\mathbb{E}_D\left[\mathbf{x}^2\right]\right\|_1}{d\left\|\mathbb{E}_D\left[\mathbf{x}^2\right]\right\|_\infty}.$$

Similar to (Cesa-Bianchi et al., 2011; Hazan & Koren, 2012), we used 10-fold cross validation to optimize the parameters for each phase. We measured the performance of

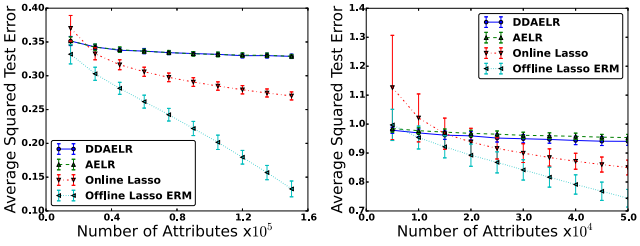


(a) $\alpha = 0$, $\rho_{\text{Ridge}} = 1$. (b) $\alpha = -0.5$, $\rho_{\text{Ridge}} = 0.91$.

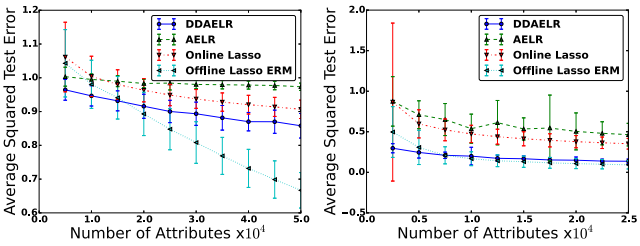


(c) $\alpha = -1$, $\rho_{\text{Ridge}} = 0.55$. (d) $\alpha = -2$, $\rho_{\text{Ridge}} = 0.05$.

Figure 1. Test error for the algorithms with $k+1 = 5$ in the Ridge scenario over simulated data with $d = 500$.



(a) $\alpha = 0$, $\rho_{\text{Lasso}} = 1$. (b) $\alpha = -0.5$, $\rho_{\text{Lasso}} = 0.086$.



(c) $\alpha = -1$, $\rho_{\text{Lasso}} = 0.014$. (d) $\alpha = -2$, $\rho_{\text{Lasso}} = 0.0033$.

Figure 2. Test error for the algorithms with $k+1 = 5$ in the Lasso scenario over simulated data with $d = 500$.

each algorithm by the average loss over the testing set, divided by the loss of the zero predictor, and defined the error bars as one standard deviation over 100 repeats of each experiment. For the two-phased algorithms, we set $m_1 = \frac{m}{10}$, $m_2 = \frac{9m}{10}$, and run the AERR/AELR algorithm during the first phase, using its result as a starting point for the second phase. Unlike the theoretical analysis, we set ϵ to 0, since the theoretical upper confidence bound is conservative, and split the attribute budget evenly between the data point estimation and the inner product estimation as we found that these improve the empirical results.

5.1. Simulated Data

We begin by studying a synthetic linear data set which allows us to control the improvement ratio in both scenarios and to demonstrate the dependence of the algorithms on them. We first defined a vector $\mathbf{u} \in \mathbb{R}^d$ (where $d = 500$) with exponentially decaying coefficients: $u_i = i^\alpha$ for some $\alpha \leq 0$ and projected it on the L_2 (L_∞) ball of radius 1 for the Ridge (Lasso) scenario, to produce the expected values of each attribute. To generate one training example, we generated independent binary variables with the corresponding expectations, and joined them into a d-dimensional vector. To generate the entire training set, we repeated the example generation process independently m times. In all these experiments, we used $k + 1 = 5$.

In the Ridge scenario, the target values were generated using a scalar product with a random weight vector from $\{-1, 1\}^d$, $\mathbf{w}^*_{\text{Ridge}}$, which itself was generated i.i.d. with $P(w^*_{\text{Ridge},i} = 1) = P(w^*_{\text{Ridge},i} = -1) = 0.5$. In the Lasso scenario, the target values were generated using a scalar product with a random sparse weight vector from $\{-1, 0, 1\}^d$, $\mathbf{w}^*_{\text{Lasso}}$, which was generated i.i.d. with $P(w^*_{\text{Lasso},i} = 1) = P(w^*_{\text{Lasso},i} = -1) = 0.15$ and $P(w^*_{\text{Lasso},i} = 0) = 0.7$.

The Ridge results appear in figure 1: In the first experiment, all the attributes have the same distribution, $\rho_{\text{Ridge}} = 1$, and the DDAERR and AERR algorithms are equivalent[5]. As ρ_{Ridge} decreases, the algorithms drift apart, and we see a significant improvement in our methods.

The Lasso results that appear in figure 2 are similar, this time with respect to $\left\|\mathbb{E}_D\left[\mathbf{x}^2\right]\right\|_1$ instead of $\left\|\mathbb{E}_D\left[\mathbf{x}^2\right]\right\|_{\frac{1}{2}}$.

5.2. MNIST Data Set

In our next set of experiments, we choose to repeat the experiments in (Cesa-Bianchi et al., 2011; Hazan & Koren, 2012) and use the MNIST data set. Each training example is a labeled 28×28 grayscale image of one hand-written digit. As in the original experiments, we focused on the classification problem of distinguishing between the "3" digits (which we labeled -1) and the "5" digits (which we labeled +1) and addressed it by regressing the labels. As in (Hazan & Koren, 2012), we used $k + 1 = 57$ attributes for each training example in the Ridge scenario and $k + 1 = 5$ attributes in the Lasso scenario. For this data set we have $d = 784$, $\rho_{\text{Ridge}} = 0.45$ and $\rho_{\text{Lasso}} = 0.2$.

[5]The small difference between the algorithms is caused by the different methods of calculating η.

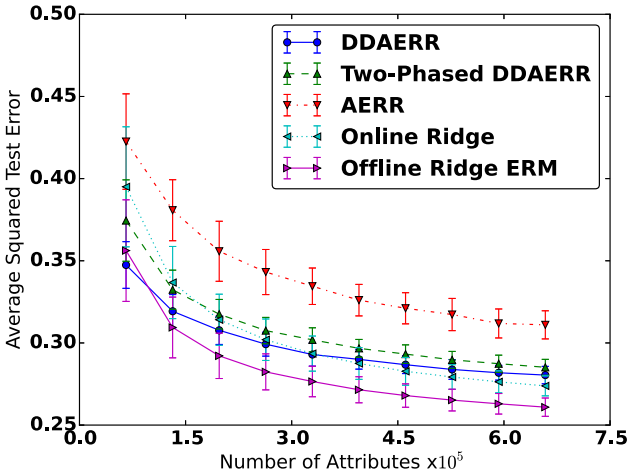


Figure 3. Test error for the algorithms with $k + 1 = 57$ in the Ridge scenario over the classification task "3" vs. "5" in the MNIST data set.

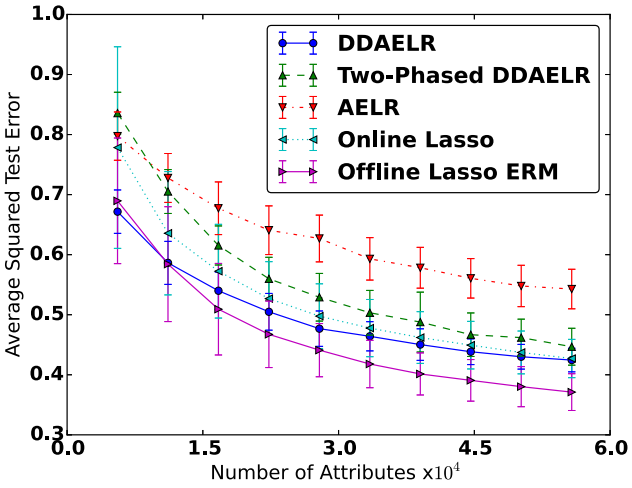


Figure 4. Test error for the algorithms with $k+1 = 5$ in the Lasso scenario over the classification task "3" vs. "5" in the MNIST data set.

The Ridge results appear in figure 3: Our DDAERR algorithm performs considerably better than the AERR algorithm, for all the training set sizes checked, in correspondence with the theory. Also, the DDAERR algorithm performs similarly to the online Ridge algorithm, and even better for a small total number of examined attributes. This suggests that at least for a small number of total attributes, our attribute efficient method is better than the full-information method. The offline Ridge algorithm is the best algorithm, because it can utilize all attributes from each example, as well as use each example more than once, unlike the attribute efficient algorithms. The performance of the Two-Phased DDAERR is between those of the AERR algorithm and the DDAERR algorithm, and converges towards the DDAERR algorithm as the number of observed attributes grows, as expected.

The Lasso results, which appear in figure 4, are similar: The DDAELR algorithm performs considerably better than the AELR algorithm, and comparable with the online Lasso algorithm, if not slightly better. It is interesting to note that the variance in the performance of the DDAELR algorithm is smaller than that of other algorithms. Also, this time it is much clearer that the Two-Phased DDAELR algorithm performs similarly to the AELR algorithm for a small amount of examined attributes, and converges to DDAELR as the number of examined attributes increases, as expected.

6. Summary and Extensions

In this paper we studied the attribute efficient local budget setting and developed efficient linear regression algorithms for the Ridge and Lasso regression scenarios. Our algorithms utilize the geometry of the data distribution, and are able to achieve distribution-dependent improvements factors for the excess risk bound over the state-of-the-art, which can be as large as $O(\sqrt{d/k})$. Moreover, under reasonable assumptions, our algorithms are the first to provably use less attributes than full-information algorithms, which is the main concern in budgeted learning.

Interestingly, our partial information algorithms can also be used to speed up learning in the full-information case: To learn a linear (Ridge) predictor in the full information case, one can use OGD, obtain a convergence rate of $\mathrm{O}\left(1/\sqrt{m}\right)$ with a cost of processing d attributes per example. However, one may also use our DDAERR algorithm by setting $k = \left\|\mathbb{E}_D\left[\mathbf{x}^2\right]\right\|_{\frac{1}{2}}$. This will result in the same convergence rate, but at the cost of processing only $k + 1$ attributes per example, which can be much faster.

There are several possible directions for future work. For example, our work focuses on learning from i.i.d. data, and it would be interesting to extend it to a non-stochastic online learning environment, where the data is possibly generated by an adversary. Another direction is to replace the dependence on the second moments of the data by a more refined notion, which also depends on the geometry of the optimal linear predictor (e.g. if it is sparse, then perhaps one can learn while sampling fewer 'irrelevant' features). Also, it would be interesting to generalize the results beyond the Ridge and Lasso scenarios to a joint framework with general norms and loss functions. Finally, proving distribution-dependent lower bounds may complement our results, or show additional room for improvements.

Acknowledgements: This research was partially supported by an Israel Science Foundation Grant (425/13) and an FP7 Marie Curie CIG grant.

References

Ben-David, Shai and Dichterman, Eli. Learning with restricted focus of attention. In *Proceedings of the sixth annual conference on Computational learning theory*, pp. 287–296. ACM, 1993.

Blackard, Jock A and Dean, Denis J. Comparative accuracies of artificial neural networks and discriminant analysis in predicting forest cover types from cartographic variables. *Computers and electronics in agriculture*, 24 (3):131–151, 1999.

Cesa-Bianchi, Nicolo, Shalev-Shwartz, Shai, and Shamir, Ohad. Efficient learning with partially observed attributes. *The Journal of Machine Learning Research*, 12:2857–2878, 2011.

Clarkson, Kenneth L, Hazan, Elad, and Woodruff, David P. Sublinear optimization for machine learning. *Journal of the ACM (JACM)*, 59(5):23, 2012.

Duchi, John, Hazan, Elad, and Singer, Yoram. Adaptive subgradient methods for online learning and stochastic optimization. *The Journal of Machine Learning Research*, 12:2121–2159, 2011.

Hazan, Elad and Koren, Tomer. Linear regression with limited observation. In *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*, pp. 807–814, 2012.

Kakade, Sham M, Sridharan, Karthik, and Tewari, Ambuj. On the complexity of linear prediction: Risk bounds, margin bounds, and regularization. In *Advances in neural information processing systems*, pp. 793–800, 2009.

Kivinen, Jyrki and Warmuth, Manfred K. Exponentiated gradient versus gradient descent for linear predictors. *Information and Computation*, 132(1):1–63, 1997.

LeCun, Yann, Bottou, Léon, Bengio, Yoshua, and Haffner, Patrick. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Madani, Omid, Lizotte, Daniel J, and Greiner, Russell. Active model selection. In *Proceedings of the 20th conference on Uncertainty in artificial intelligence*, pp. 357–365. AUAI Press, 2004.

Zolghadr, Navid, Bartók, Gábor, Greiner, Russell, György, András, and Szepesvári, Csaba. Online learning with costly features and labels. In *Advances in Neural Information Processing Systems*, pp. 1241–1249, 2013.