Journal of Machine Learning Research 12 (2011) 2857-2878 Submitted 4/11; Revised 9/11; Published 10/11

Efficient Learning with Partially Observed Attributes*

Nicolò Cesa-Bianchi NICOLO.CESA-BIANCHI@UNIMI.IT
DSI, Università degli Studi di Milano
via Comelico, 39
20135 Milano, Italy

Shai Shalev-Shwartz SHAIS@CS.HUJI.AC.IL
The Hebrew University
Givat Ram, Jerusalem 91904, Israel

Ohad Shamir OHADSH@MICROSOFT.COM
Microsoft Research
One Memorial Drive
Cambridge, MA 02142, USA

Editor: Russ Greiner

Abstract

We investigate three variants of budgeted learning, a setting in which the learner is allowed to access a limited number of attributes from training or test examples. In the "local budget" setting, where a constraint is imposed on the number of available attributes per training example, we design and analyze an efficient algorithm for learning linear predictors that actively samples the attributes of each training instance. Our analysis bounds the number of additional examples sufficient to compensate for the lack of full information on the training set. This result is complemented by a general lower bound for the easier "global budget" setting, where it is only the overall number of accessible training attributes that is being constrained. In the third, "prediction on a budget" setting, when the constraint is on the number of available attributes per test example, we show that there are cases in which there exists a linear predictor with zero error but it is statistically impossible to achieve arbitrary accuracy without full information on test examples. Finally, we run simple experiments on a digit recognition problem that reveal that our algorithm has a good performance against both partial information and full information baselines.

Keywords: budgeted learning, statistical learning, linear predictors, learning with partial information, learning theory

1. Introduction

Consider the problem of predicting whether a person has some disease based on medical tests. In principle, we may draw a sample of the population, perform a large number of medical tests on each person in the sample, and use this information to train a classifier. In many situations, however, this approach is unrealistic. First, patients participating in the experiment are generally not willing to go through a large number of medical tests. Second, each test has some associated cost, and we typically have a budget on the amount of money to spend for collecting the training information. This scenario, where there is a hard constraint on the number of training attributes the

*. A short version of this paper has been presented in ICML 2010.

learner has access to, is known as *budgeted learning*.[1] Note that the constraint on the number of training attributes may be local (no single participant is willing to undergo many tests) or global (the overall number of tests that can be performed is limited). In a different but related budgeted learning setting, the system may be facing a restriction on the number of attributes that can be viewed at test time. This may happen, for example, in a search engine, where a ranking of web pages must be generated for each incoming user query and there is no time to evaluate a large number of attributes to answer the query.

We may thus distinguish three basic budgeted learning settings:

- **Local Budget Constraint:** The learner has access to at most k attributes of each individual example, where k is a parameter of the problem. The learner has the freedom to actively choose *which* of the attributes is revealed, as long as at most k of them will be given.

- **Global Budget Constraint:** The total number of training attributes the learner is allowed to see is bounded by k. As in the local budget constraint setting, the learner has the freedom to actively choose which of the attributes is revealed. In contrast to the local budget constraint setting, the learner can choose to access more than k/m attributes from specific examples (where m is the overall number of examples) as long as the global number of attributes is bounded by k.

- **Prediction on a budget:** The learner receives the entire training set, however, at test time, the predictor can see at most k attributes of each instance and then must form a prediction. The predictor is allowed to actively choose which of the attributes is revealed.

In this paper we focus on budgeted linear regression, and prove negative and positive learning results in the three abovementioned settings. Our first result shows that, under a *global* budget constraint, no algorithm can learn a general d-dimensional linear predictor while observing less than $\Omega(d)$ attributes at training time. This is complemented by the following positive result: we show an efficient algorithm for learning under a given *local* budget constraint of $2k$ attributes per example, for any $k \geq 1$. The algorithm actively picks which attributes to observe in each example in a randomized way depending on past observed attributes, and constructs a "noisy" version of *all* attributes. Intuitively, we can still learn despite the error of this estimate because instead of receiving the exact value of each individual example in a small set it suffices to get noisy estimations of many examples. We show that the overall number of attributes our algorithm needs to learn a regressor is at most a factor of d bigger than that used by standard regression algorithms that view all the attributes of each example. Ignoring logarithmic factors, the same gap of d exists when the attribute bound of our algorithm is specialized to the choice of parameters that is used to prove the abovementioned $\Omega(d)$ lower bound under the global budget constraint.

In the prediction on a budget setting, we prove that in general it is not possible (even with an infinite amount of training examples) to build an active classifier that uses at most two attributes of each example at test time, and whose error will be smaller than a constant. This in contrast with the local budget setting, where it is possible to learn a consistent predictor by accessing at most two attributes of each example at training time.

1. See, for example, `webdocs.cs.ualberta.ca/~greiner/BudgetedLearning/`.

2. Related Work

The notion of budgeted learning is typically identified with the "global budget" and "prediction on a budget" settings—see, for example, Deng et al. (2007), Kapoor and Greiner (2005a,b) and Greiner et al. (2002) and references therein. The more restrictive "local budget" setting has been first proposed in Ben-David and Dichterman (1998) under the name of "learning with restricted focus of attention". Ben-David and Dichterman (1998) considered binary classification and showed learnability of several hypothesis classes in this model, like k-DNF and axis-aligned rectangles. However, to the best of our knowledge, no efficient algorithm for the class of linear predictors has been so far proposed.[2]

Our algorithm for the local budget setting actively chooses which attributes to observe for each example. Similarly to the heuristics of Deng et al. (2007), we borrow ideas from the adversarial multi-armed bandit problem (Auer et al., 2003; Cesa-Bianchi and Lugosi, 2006). However, our algorithm is guaranteed to be attribute efficient, comes with finite sample generalization bounds, and is provably competitive with algorithms which enjoy full access to the data. A related but different setting is multi-armed bandit on a global budget—see, for example, Guha and Munagala (2007) and Madani et al. (2004). There one learns the single best arm rather than the best linear combination of many attributes, as we do here. Similar protocols were also studied in the context of active learning (Cohn et al., 1994; Balcan et al., 2006; Hanneke, 2007, 2009; Beygelzimer et al., 2009), where the learner can ask for the target associated with specific examples.

Finally, our technique is reminiscent of methods used in the compressed learning framework (Calderbank et al., 2009; Zhou et al., 2009), where data is accessed via a small set of random linear measurements. Unlike compressed learning, where learners are both trained and evaluated in the compressed domain, our techniques are mainly designed for a scenario in which only the access to training data is restricted.

We note that a recent follow-up work (Hazan and Koren, 2011) present 1-norm and 2-norm based algorithms for our local budget setting, whose theoretical guarantees improve on those presented in this paper, and match our lower bound to within logarithmic factors.

3. Linear Regression

We consider linear regression problems where each example is an instance-target pair, $(x,y) \in \mathbb{R}^d \times \mathbb{R}$. We refer to x as a vector of attributes. Throughout the paper we assume that $\|x\|_\infty \leq 1$ and $|y| \leq B$. The goal of the learner is to find a linear predictor $x \mapsto \langle w,x \rangle$. In the rest of the paper, we use the term predictor to denote the vector $w \in \mathbb{R}^d$. The performance of a predictor w on an instance-target pair, $(x,y) \in \mathbb{R}^d \times \mathbb{R}$, is measured by a loss function $\ell(\langle w,x \rangle, y)$. For simplicity, we focus on the squared loss function, $\ell(a,b) = (a-b)^2$, and briefly mention other loss functions in Section 8. Following the standard framework of statistical learning (Haussler, 1992; Devroye et al., 1996; Vapnik, 1998), we model the environment as a joint distribution $\mathcal{D}$ over the set of instance-target pairs, $\mathbb{R}^d \times \mathbb{R}$. The goal of the learner is to find a predictor with low risk, defined as the expected loss

$$L_{\mathcal{D}}(w) \stackrel{\text{def}}{=} \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}} \left[\ell(\langle w,x \rangle, y) \right] .$$

2. Ben-David and Dichterman (1998) do describe learnability results for similar classes but only under the restricted family of product distributions.

Since the distribution $\mathcal{D}$ is unknown, the learner relies on a training set of m examples $S = \{(x_1, y_1), \ldots, (x_m, y_m)\}$, which are assumed to be sampled i.i.d. from $\mathcal{D}$. We denote the training loss by

$$L_S(w) \stackrel{\text{def}}{=} \frac{1}{m} \sum_{i=1}^{m} (\langle w, x_i \rangle - y_i)^2 \ .$$

4. Impossibility Results

Our first result states that any budget learning algorithm (local or global) needs in general a budget of $\Omega(d)$ attributes for learning a d-dimensional linear predictor.

Theorem 1 *For any $d \geq 4$ and $\varepsilon \in \left(0, \frac{1}{16}\right)$, there exists a distribution $\mathcal{D}$ over $\{-1,+1\}^d \times \{-1,+1\}$ and a weight vector $w^\star \in \mathbb{R}^d$, with $\|w^\star\|_0 = 1$ and $\|w^\star\|_2 = \|w^\star\|_1 = 2\sqrt{\varepsilon}$, such that any learning algorithm must see at least*

$$k \geq \frac{1}{2} \left\lfloor \frac{d}{96\varepsilon} \right\rfloor$$

attributes in order to learn a linear predictor w such that $L_{\mathcal{D}}(w) - L_{\mathcal{D}}(w^\star) < \varepsilon$.

The proof is given in the Appendix. In Section 6 we prove that under the same assumptions as those of Theorem 1, it is possible to learn a predictor using a local budget of two attributes per example and using a total of $\widetilde{O}(d^2)$ training examples. Thus, ignoring logarithmic factors hidden in the $\widetilde{O}$ notation, we have a multiplicative gap of d between the lower bound and the upper bound.

Next, we consider the prediction on a budget setting. Greiner et al. (2002) studied this setting and showed positive results regarding (agnostic) PAC-learning of *k-active* predictors. A k-active predictor is restricted to use at most k attributes per test example x, where the choice of the i-th attribute of x may depend on the values of the $i-1$ attributes of x that have been already observed. Greiner et al. (2002) show that it is possible to learn a k-active predictor from training examples whose performance is slightly worse than that of the best k-active predictor. But, how good are the predictions of the best k-active predictor? We now show that even in simple cases in which there exists a linear predictor $w^\star$ with $L_{\mathcal{D}}(w^\star) = 0$, the risk of the best k-active predictor can be high. The following theorem indeed shows that if the only constraint on $w^\star$ is bounded ℓ_2 norm, then the risk can be as high as $1 - \frac{k}{d}$. We use the notation $L_{\mathcal{D}}(A)$ to denote the expected loss of the k-active predictor A on a test example.

Theorem 2 *There exists a weight vector $w^\star \in \mathbb{R}^d$ and a distribution $\mathcal{D}$ such that $\|w^\star\|_2 = 1$ and $L_{\mathcal{D}}(w^\star) = 0$, while any k-active predictor A must have $L_{\mathcal{D}}(A) \geq 1 - \frac{k}{d}$.*

Note that the risk of the constant prediction of zero is 1. Therefore, the theorem tells us that no active predictor can get an improvement over the naive predictor of more than $\frac{k}{d}$.

Proof For any $d > k$ let $w^\star = \left(1/\sqrt{d}, \ldots, 1/\sqrt{d}\right)$. Let $x \in \{\pm 1\}^d$ be distributed uniformly at random and y is determined deterministically to be $\langle w^\star, x \rangle$. Then, $L_{\mathcal{D}}(w^\star) = 0$ and $\|w^\star\|_2 = 1$. Without loss of generality, suppose the k-active predictor asks for the first k attributes of a test example and forms its prediction to be $\hat{y}$. Since the generation of attributes is independent, we have that the value of $x_{k+1}, \ldots, x_d$ does not depend neither on $x_1, \ldots, x_k$ nor on $\hat{y}$. Using this and the fact that $\mathbb{E}[x_j] = 0$ for

all j we therefore obtain

$$\begin{aligned}
\mathbb{E}\left[(\hat{y}-\langle w^\star,x\rangle)^2\right] &= \mathbb{E}\left[\left(\hat{y}-\sum_{i=1}^{k} w_i^\star x_i - \sum_{j=k+1}^{d} w_j^\star x_j\right)^2\right] \\
&= \mathbb{E}\left[\left(\hat{y}-\sum_{i=1}^{k} w_i^\star x_i\right)^2\right] + \sum_{j=k+1}^{d} (w_j^\star)^2\,\mathbb{E}[x_j^2] \\
&\quad + 2\hat{y}\sum_{j=k+1}^{d} w_j^\star\,\mathbb{E}[x_j] - 2\sum_{i=1}^{k}\sum_{j=k+1}^{d} w_i^\star w_j^\star\,\mathbb{E}[x_i]\,\mathbb{E}[x_j] \\
&= \mathbb{E}\left[\left(\hat{y}-\sum_{i=1}^{k} w_i^\star x_i\right)^2\right] + \sum_{i>k}(w_i^\star)^2\,\mathbb{E}[x_i^2] + 0 \\
&\geq 0 + \frac{d-k}{d} = 1-\frac{k}{d}
\end{aligned}$$

which concludes our proof. ■

It is well known that a low 1-norm of $w^\star$ encourages sparsity of the learned predictor, which naturally helps in designing active predictors. The following theorem shows that even if we restrict $w^\star$ to have $\|w^\star\|_1 = 1$, $L_{\mathcal{D}}(w^\star) = 0$, and $\|w^\star\|_0 > k$, we still have that the risk of the best k-active predictor can be non-vanishing.

Theorem 3 *There exists a weight vector $w^\star \in \mathbb{R}^d$ and a distribution $\mathcal{D}$ such that $\|w^\star\|_1 = 1$, $L_{\mathcal{D}}(w^\star) = 0$, and $\|w^\star\|_0 = ck$ (for $c > 1$) such that any k-active predictor A must have $L_{\mathcal{D}}(A) \geq \left(1-\frac{1}{c}\right)\frac{1}{ck}$.*

For example, if in the theorem above we choose $c = 2$, then $\|w^\star\|_0 = 2k$ and $L_{\mathcal{D}}(A) \geq \frac{1}{4k}$. If we choose instead $c = \frac{k+1}{k}$, then $\|w^\star\|_0 = k+1$ and $L_{\mathcal{D}}(A) \geq \frac{1}{(k+1)^2}$. Note that if $\|w^\star\|_0 \leq k$ there is a trivial way to predict on a budget of k attributes by always querying the attributes corresponding to the non-zero elements of $w^\star$.

Proof Let

$$w^\star = \big(\underbrace{\tfrac{1}{ck},\ldots,\tfrac{1}{ck}}_{ck \text{ components}},0,\ldots,0\big)$$

and, similarly to the proof of Theorem2, let $x \in \{\pm 1\}^d$ be distributed uniformly at random and let y be determined deterministically to be $\langle w^\star,x\rangle$. Then, $L_{\mathcal{D}}(w^\star) = 0$, $\|w^\star\|_1 = 1$, and $\|w^\star\|_0 = ck$. Without loss of generality, suppose the k-active predictor asks for the first $k < ck$ attributes of a test example and form its prediction to be $\hat{y}$. Again similarly to the proof of Theorem2, since the generation of attributes is independent, we have that the value of $x_{k+1},\ldots,x_d$ does not depend on

$x_1,\ldots,x_k$, and on $\hat{y}$. Therefore,

$$\begin{aligned}\mathbb{E}\left[(\hat{y}-\langle w^\star,x\rangle)^2\right] &= \mathbb{E}\left[\left(\hat{y}-\sum_{i=1}^{k} w_i^\star x_i\right)^2\right]+\sum_{i>k}(w_i^\star)^2\,\mathbb{E}[x_i^2]\\ &\geq 0+\frac{ck-k}{(ck)^2}\\ &= \frac{c-1}{c^2k}=\left(1-\frac{1}{c}\right)\frac{1}{ck}\end{aligned}$$

which concludes our proof. ■

These negative results highlight an interesting phenomenon: in Section 6 we show that one can learn an arbitrarily accurate predictor w with a local budget of $k=2$. However, here we show that even if we know the optimal $w^\star$, we might not be able to accurately predict a new partially observed example unless k is very large. Therefore, at least in the worst-case sense, learning on a budget is much easier than predicting on a budget.

5. Local Budget Constraint: A Baseline Algorithm

In this section we describe a straightforward adaptation of Lasso (Tibshirani, 1996) to the local budget setting. This adaptation is based on a direct nonadaptive estimate of the loss function. In Section 6 we describe a more effective approach, which combines a stochastic gradient descent algorithm called Pegasos (Shalev-Shwartz et al., 2007) with the adaptive sampling of attributes to estimate the gradient of the loss at each step.

A popular approach for learning a linear regressor is to minimize the empirical loss on the training set plus a regularization term, which often takes the form of a norm of the predictor w. For example, in ridge regression the regularization term is $\|w\|_2^2$ and in Lasso the regularization term is $\|w\|_1$. Instead of regularization, we can include a constraint of the form $\|w\|_1 \leq B$ or $\|w\|_2 \leq B$. Modulo an appropriate choice of the parameters, the regularization form is equivalent to the constraint form. In the constraint form, the predictor is a solution to the following optimization problem

$$\begin{aligned}\min_{w\in\mathbb{R}^d} \quad & \frac{1}{|S|}\sum_{(x,y)\in S}\left(\langle w,x\rangle - y\right)^2\\ \text{s.t.} \quad & \|w\|_p \leq B\end{aligned} \tag{1}$$

where $S=\{(x_1,y_1),\ldots,(x_m,y_m)\}$ is a training set of m examples, B is the regularization parameter, and p is 1 for Lasso and 2 for ridge regression.

We start with a standard risk bound for constrained predictors.

Lemma 4 *Let $\mathcal{D}$ be a distribution on pairs $(x,y)\in\mathbb{R}^d\times\mathbb{R}$ such that $\|x\|_\infty\leq 1$ and $|y|\leq B$ holds with probability one. Then there exists a constant $c>0$ such that*

$$\max_{w:\,\|w\|_1\leq B}\left|L_S(w)-L_{\mathcal{D}}(w)\right| = cB^2\sqrt{\frac{1}{m}\ln\frac{d}{\delta}}\,.$$

holds with probability at least $1-\delta$ with respect to the random draw of the training set S of size m from $\mathcal{D}$.

Proof We apply the following Rademacher bound (Kakade et al., 2008)

$$\left|L_S(w) - L_{\mathcal{D}}(w)\right| \leq L_{\max} B \sqrt{\frac{2}{m} \ln 2d} + \ell_{\max} \sqrt{\frac{1}{2m} \ln \frac{2}{\delta}}$$

that holds with probability at least $1-\delta$ for all $w \in \mathbb{R}^d$ such that $\|w\|_1 \leq B$, where $L_{\max}$ bounds the Lipschitz constant for the square loss from above, and $\ell_{\max}$ bounds the square loss from above. The result then follows by observing that $\left|(a-y)^2 - (b-y)^2\right| \leq |a-b|\,|a+b-2y|$. Hence, $L_{\max} \leq \max_{a,b,y} |a+b-2y| = 4B$ where both a and b are of the form $\langle w, x\rangle$, and we used the fact $\left|\langle w, x\rangle\right| \leq B$ (recall that $\|x\|_\infty \leq 1$) together with the assumption $|y| \leq B$. Similarly, under the same assumptions, $\ell_{\max} = \max_{a,y} (a-y)^2 = 4B^2$. ■

This immediately leads to the following risk bound for Lasso.

Corollary 5 *If $\widehat{w}$ is a minimizer of (1) with $p = 1$, then there exists a constant $c > 0$ such that, under the same assumptions as Lemma 4,*

$$L_{\mathcal{D}}(\widehat{w}) \leq \min_{w:\, \|w\|_1 \leq B} L_D(w) + c B^2 \sqrt{\frac{1}{m} \ln \frac{d}{\delta}} \tag{2}$$

holds with probability at least $1-\delta$ over the random draw of the training set S of size m from $\mathcal{D}$.

To adapt Lasso to the partial information case, we first rewrite the squared loss as follows:

$$\left(\langle w, x\rangle - y\right)^2 = w^\top x x^\top w - 2y x^\top w + y^2$$

where w, x are column vectors and $w^\top, x^\top$ are their corresponding transpose (i.e., row vectors). Next, we estimate the matrix $xx^\top$ and the vector x using the partial information we have, and then we solve the optimization problem given in (1) with the estimated values of $xx^\top$ and x. To estimate the vector x we can pick an index i uniformly at random from $[d] = \{1, \ldots, d\}$ and define the estimation to be a vector v such that

$$v_r = \begin{cases} d\, x_r & \text{if } r = i \\ 0 & \text{else} \end{cases}. \tag{3}$$

It is easy to verify that v is an unbiased estimate of x, namely, $\mathbb{E}[v] = x$ where expectation is with respect to the choice of the index i. To estimate the matrix $xx^\top$ we could pick two indices i, j independently and uniformly at random from $[d]$, and define the estimation to be a matrix with all zeros except $d^2 x_i x_j$ in the (i, j) entry. However, this yields a non-symmetric matrix which will make our optimization problem with the estimated matrix non-convex. To overcome this obstacle, we symmetrize the matrix by adding its transpose and dividing by 2. This sampling process can be easily generalized to the case where $k > 1$ attributes can be seen. The resulting baseline procedure[3] is given in Algorithm 1.

The following theorem shows that similar to Lasso, the Baseline algorithm is competitive with the optimal linear predictor with a bounded 1-norm.

3. We note that an even simpler approach is to arbitrarily assume that the correlation matrix is the identity matrix and then the solution to the loss minimization problem is simply the averaged vector, $w = \sum_{(x,y)\in S} y x$. In that case, we can simply replace x by its estimated vector as defined in (3). While this naive approach can work on very simple classification tasks, it will perform poorly on realistic data sets, in which the correlation matrix is not likely to be identity. Indeed, in our experiments with the MNIST data set, we found out that this approach performed poorly relatively to the algorithms proposed in this paper.

ALGORITHM: Baseline(S,k)
INPUT: Training set S of size m, local budget $k \geq 2$ (with k even)
INITIALIZE: $\bar{A} = 0 \in \mathbb{R}^{d\times d}$; $\bar{v} = 0 \in \mathbb{R}^d$; $\bar{y} = 0$

for each $(x,y) \in S$
  $v = 0 \in \mathbb{R}^d$; $A = 0 \in \mathbb{R}^{d\times d}$
  Choose a set C of k entries from $[d]$, uniformly without replacement
  for each $c \in C$
    $v_c = v_c + \frac{d}{k} x_c$
  Randomly split C into two sets I, J of size $k/2$ each
  for each $(i,j) \in I \times J$
    $A_{i,j} = A_{i,j} + 2\left(\frac{d}{k}\right)^2 x_i x_j \quad ; \quad A_{j,i} = A_{j,i} + 2\left(\frac{d}{k}\right)^2 x_i x_j$
  end
  $\bar{A} = \bar{A} + \frac{A}{m} \quad ; \quad \bar{v} = \bar{v} + 2y\frac{v}{m} \quad ; \quad \bar{y} = \bar{y} + \frac{y^2}{m}$
end
Let $\tilde{L}_S(w) = w^\top \bar{A} w + w^\top \bar{v} + \bar{y}$
OUTPUT: $\widehat{w} = \operatorname*{argmin}_{w:\|w\|_1 \leq B} \tilde{L}_S(w)$

Figure 1: An adaptation of Lasso to the local budget setting, where the learner can view at most k attributes of each training example. The predictive performance of this algorithm is analyzed in Theorem 6.

Theorem 6 *Let $\mathcal{D}$ be a distribution on pairs $(x,y) \in \mathbb{R}^d \times \mathbb{R}$ such that $\|x\|_\infty \leq 1$ and $|y| \leq B$ with probability one. Let $\hat{w}$ be the output of Baseline(S,k), where $|S| = m$. Then there exists a constant $c > 0$ such that*

$$L_{\mathcal{D}}(\hat{w}) \leq \min_{w:\|w\|_1 \leq B} L_D(w) + c\left(\frac{dB}{k}\right)^2 \sqrt{\frac{1}{m}\ln\frac{d}{\delta}}$$

holds with probability of at least $1-\delta$ over the random draw of the training set S from $\mathcal{D}$ and the algorithm's own randomization.

The above theorem tells us that for a sufficiently large training set we can find a very good predictor. Put another way, a large number of examples can compensate for the lack of full information on each individual example. In particular, to overcome the extra factor $(d/k)^2$ in the bound, which does not appear in the full information bound given in (2), we need to increase m by a factor of $(d/k)^4$. In the next subsection, we describe a better, adaptive procedure for the partial information case.

In view of proving Theorem 6, we first show that sampling k elements without replacements and then averaging the result has the same expectation as sampling just once.

Lemma 7 *Let C be a set of n elements and let $f : C \to \mathbb{R}$ be an arbitrary function. Let $\mathcal{C}_k = \{C' \subset C : |C'| = k\}$ and let U be the uniform distribution over $\mathcal{C}_k$. Then*

$$\mathop{\mathbb{E}}_{C' \sim U} \left[\frac{1}{k} \sum_{c' \in C'} f(c') \right] = \frac{1}{n} \sum_{c \in C} f(c) \,.$$

Proof We have

$$\begin{aligned}
\mathop{\mathbb{E}}_{C' \sim U(} \left[\frac{1}{k} \sum_{c' \in C'} f(c') \right] &= \frac{1}{\binom{n}{k}} \sum_{C' \in \mathcal{C}_k} \frac{1}{k} \sum_{c' \in C'} f(c') \\
&= \frac{1}{k\binom{n}{k}} \sum_{c \in C} f(c) \left| \{C' \in \mathcal{C}_k : c' \in C'\} \right| \\
&= \frac{\binom{n-1}{k-1}}{k\binom{n}{k}} \sum_{c \in C} f(c) \\
&= \frac{1}{n} \sum_{c \in C} f(c)
\end{aligned}$$

and this concludes the proof. ■

We now show that the estimation matrix constructed by the Baseline algorithm is likely to be close to the true correlation matrix over the training set.

Lemma 8 *Let A_t be the matrix constructed at iteration t of the Baseline algorithm and note that $\bar{A} = \frac{1}{m} \sum_{t=1}^{m} A_t$. Let $X = \frac{1}{m} \sum_{t=1}^{m} x_t x_t^\top$. Then, with probability of at least $1 - \delta$ over the algorithm's own randomness we have that*

$$\left| \bar{A}_{r,s} - X_{r,s} \right| \leq \left(\frac{d}{k} \right)^2 \sqrt{\frac{8}{m} \ln \left(\frac{2d^2}{\delta} \right)} \qquad r, s = 1, \ldots, d.$$

Proof Based on Lemma 7, it is easy to verify that $\mathbb{E}[A_t] = x_t^\top x_t$. Additionally, since we sample without replacements, each element of A_t is in $\left[-2\left(\frac{d}{k}\right)^2, 2\left(\frac{d}{k}\right)^2 \right]$ because we assume $\|x_t\|_\infty \leq 1$. Therefore, we can apply Hoeffding's inequality on each element of $\bar{A}$ and obtain that

$$\mathbb{P}\left[\left| \bar{A}_{r,s} - X_{r,s} \right| > \varepsilon \right] \leq 2 \exp\left(-\frac{m\varepsilon^2}{8} \left(\frac{k}{d} \right)^4 \right) \,.$$

Combining the above with the union bound we obtain that

$$\mathbb{P}\left[\exists (r,s) : \left| \bar{A}_{r,s} - X_{r,s} \right| > \varepsilon \right] \leq 2d^2 \exp\left(-\frac{m\varepsilon^2}{8} \left(\frac{k}{d} \right)^4 \right) \,.$$

Setting the right-hand side of the above to δ and rearranging terms concludes the proof. ■

Next, we show that the estimate of the linear part of the objective function is also likely to be accurate.

Lemma 9 *Let v_t be the vector constructed at iteration t of the Baseline algorithm and note that $\overline{v} = \frac{1}{m}\sum_{t=1}^{m} 2y_t v_t$. Let $\overline{x} = \frac{1}{m}\sum_{t=1}^{m} 2y_t x_t$. Then, with probability at least $1-\delta$ over the algorithm's own randomness we have that*

$$\|\overline{v} - \overline{x}\|_\infty \leq \frac{dB}{k}\sqrt{\frac{8}{m}\ln\left(\frac{2d}{\delta}\right)}\,.$$

Proof Based on Lemma 7, it is easy to verify that $\mathbb{E}[2y_t v_t] = 2y_t x_t$. Additionally, since we sample k elements without replacement, each element of v_t is in $\left[-\frac{d}{k}, \frac{d}{k}\right]$ (because we assume $\|x_t\|_\infty \leq 1$) and thus each element of $2y_t v_t$ is in $\left[-\frac{2dB}{k}, \frac{2dB}{k}\right]$ (because we assume that $|y_t| \leq B$). Therefore, we can apply Hoeffding's inequality on each element of $\overline{v}$ and obtain that

$$\mathbb{P}\left[\left|\overline{v}_r - \overline{x}_r\right| > \varepsilon\right] \leq 2\exp\left(-\frac{m\varepsilon^2}{8}\left(\frac{k}{dB}\right)^2\right)\,.$$

Combining the above with the union bound we obtain that

$$\mathbb{P}\left[\exists(r,s) : \left|\overline{A}_{r,s} - X_{r,s}\right| > \varepsilon\right] \leq 2d\exp\left(-\frac{m\varepsilon^2}{8}\left(\frac{k}{dB}\right)^2\right)\,.$$

Setting the right-hand side of the above to δ and rearranging terms concludes proof. ■

Next, we show that the estimated training loss

$$\tilde{L}_S(w) = w^\top \overline{A} w + w^\top \overline{v} + \overline{y}$$

computed by the Baseline algorithm is close to the true training loss.

Lemma 10 *With probability greater than $1-\delta$ over the Baseline algorithm's own randomization, for all w such that $\|w\|_1 \leq B$,*

$$\left|\tilde{L}_S(w) - L_S(w)\right| \leq \left(\frac{Bd}{k}\right)^2\sqrt{\frac{32}{m}\ln\left(\frac{2d^2}{\delta}\right)}\,.$$

Proof Using twice Hölder's inequality and Lemma 8 we get

$$\begin{aligned}\left|w^\top(\overline{A} - X)w\right| &\leq \|w\|_1\left\|(\overline{A} - X)w\right\|_\infty \leq \|w\|_1^2 \max_{r,s=1,\ldots,d}\left|(\overline{A} - X)_{r,s}\right| \\ &\leq \left(\frac{Bd}{k}\right)^2\sqrt{\frac{8}{m}\ln\left(\frac{2d^2}{\delta}\right)}\,. \qquad (4)\end{aligned}$$

Similarly, using Hölder's inequality and Lemma 9 we also get

$$\left|w^\top(\overline{v} - \overline{x})\right| \leq \frac{B^2 d}{k}\sqrt{\frac{8}{m}\ln\left(\frac{2d}{\delta}\right)}\,. \qquad (5)$$

Using the triangle inequality, (4)–(5), and the union bound we finally obtain

$$\begin{aligned}\left|\tilde{L}_S(w) - L_S(w)\right| &= \left|w^\top \bar{A} w + w^\top \bar{v} + \bar{y} - w^\top X w - w^\top \bar{x} - \bar{y}\right| \\ &\leq \left|w^\top (\bar{A} - X) w\right| + \left|w^\top (\bar{v} - \bar{x})\right| \\ &\leq \left(\frac{Bd}{k}\right)^2 \sqrt{\frac{8}{m} \ln\left(\frac{2d^2}{\delta}\right)} + \frac{B^2 d}{k} \sqrt{\frac{8}{m} \ln\left(\frac{2d}{\delta}\right)}\end{aligned}$$

which upon slight simplifications concludes the proof. ■

We are now ready to prove Theorem 6.

Proof (of Theorem 6) Lemma 4 states that with probability greater than $1-\delta$ over the random draw of a training set S of m examples, for all w such that $\|w\|_1 \leq B$, we have that

$$\left|L_S(w) - L_{\mathcal{D}}(w)\right| = c' B^2 \sqrt{\frac{1}{m} \ln \frac{d}{\delta}}$$

for some $c' > 0$. Combining the above with Lemma 10, we obtain that for some $c > 0$, with probability at least $1-\delta$ over both the random draw of the training set and the algorithm's own randomization,

$$\left|L_{\mathcal{D}}(w) - \tilde{L}_S(w)\right| \leq \left|L_{\mathcal{D}}(w) - L_S(w)\right| + \left|L_S(w) - \tilde{L}_S(w)\right| \leq c \left(\frac{dB}{k}\right)^2 \sqrt{\frac{1}{m} \ln \frac{d}{\delta}}$$

for all w such that $\|w\|_1 \leq B$. The proof of Theorem 6 follows since the Baseline algorithm minimizes $\tilde{L}_S(w)$. ■

6. Gradient-Based Attribute Efficient Regression

In this section, by avoiding the estimation of the matrix $xx^\top$, we significantly decrease the number of additional examples sufficient for learning with k attributes per training example. To do so, we do not try to estimate the loss function but rather to estimate the *gradient* $\nabla \ell(w) = 2\big(\langle w, x\rangle - y\big)x$, with respect to w, of the squared loss function $\ell(w) = \big(\langle w, x\rangle - y\big)^2$. Each vector w defines a probability distribution P over $[d]$ by letting $P(i) = |w_i| / \|w\|_1$. We can estimate the gradient using an even number $k \geq 2$ of attributes as follows. First, we randomly pick a subset $i_1, \ldots, i_{k/2}$ from $[d]$ according to the uniform distribution over the $k/2$-subsets in $[d]$. Based on this, we estimate the vector x via

$$v = \frac{2}{k} d \sum_{s=1}^{k/2} x_{i_s} e_{i_s} \tag{6}$$

where e_j is the j-th element of the canonical basis of $\mathbb{R}^d$. Second, we randomly pick $j_1, \ldots, j_{k/2}$ from $[d]$ without replacement according to the distribution defined by w. Based on this, we estimate the term $\langle w, x\rangle$ by

$$\hat{y} = \frac{2}{k} \|w\|_1 \sum_{s=1}^{k/2} \operatorname{sgn}\big(w_{j_s}\big) x_{j_s} e_{j_s} \,. \tag{7}$$

This allows us to obtain an unbiased estimate of the gradient, as stated by the following simple result.

Lemma 11 *Fix any $w, x \in \mathbb{R}^d$ and $y \in \mathbb{R}$ and let $\ell(w) = \big(\langle w,x\rangle - y\big)$ be the square loss. Then the estimate*

$$\widetilde{\nabla}\ell(w) = 2\big(\hat{y} - y\big)v \tag{8}$$

satisfies $\mathbb{E}\,\widetilde{\nabla}\ell(w) = 2\big(\langle w,x\rangle - y\big)x = \nabla\ell(w)$.

Proof Since $\mathbb{E}[d\,x_j\,e_j] = x$ for a random $j \in [d]$, Lemma 7 immediately implies that $\mathbb{E}[v] = x$. Moreover, it is easy to see that $\mathbb{E}\big[\|w\|_1 \operatorname{sgn}(w_i)\,x_i\,e_i\big] = \langle w,x\rangle$ when i is drawn with probability $P(i) = |w_i|\big/\|w\|_1$. Hence $\mathbb{E}[\hat{y}] = \langle w,x\rangle$. The proof is concluded by noting that $i_1,\ldots,i_{k/2}$ are drawn independently from $j_1,\ldots,j_{k/2}$. ■

The advantage of the above approach over the loss based approach we took before is that the magnitude of each element of the gradient estimate is order of $d\,\|w\|_1$. This is in contrast to what we had for the loss based approach, where the magnitude of each element of the matrix A was order of d^2. In many situations, the 1-norm of a good predictor is significantly smaller than d and in these cases the gradient based estimate is better than the loss based estimate. However, while in the previous approach our estimation did not depend on a specific w, now the estimation depends on w. We therefore need an iterative learning method in which at each iteration we use the gradient of the loss function on an individual example. Luckily, the stochastic gradient descent approach conveniently fits our needs.

Concretely, below we describe a variant of the Pegasos algorithm (Shalev-Shwartz et al., 2007) for learning linear regressors. Pegasos tries to minimize the regularized risk

$$\min_w \frac{\lambda}{2}\|w\|_2^2 \quad + \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}}\Big[\big(\langle w,x\rangle - y\big)^2\Big]\,. \tag{9}$$

Of course, the distribution $\mathcal{D}$ is unknown, and therefore we cannot hope to solve the above problem exactly. Instead, Pegasos finds a sequence of weight vectors that (on average) converge to the solution of (9). We start with the all zeros vector $w = 0 \in \mathbb{R}^d$. Then, at each iteration Pegasos picks the next example in the training set (which is equivalent to sampling a fresh example according to $\mathcal{D}$) and calculates the gradient of the regularized loss

$$g(w) = \frac{\lambda}{2}\|w\|_2^2 + \big(\langle w,x\rangle - y\big)^2$$

for this example with respect to the current weight vector w. This gradient is simply $\nabla g(w) = \lambda w + \nabla\ell(w)$, where $\nabla\ell(w) = 2\big(\langle w,x\rangle - y\big)x$. Finally, Pegasos updates the predictor according to the gradient descent rule $w \leftarrow w - \frac{1}{\lambda t}\nabla g(w)$ where t is the current iteration number. This can be rewritten as $w \leftarrow \big(1 - \frac{1}{t}\big)w - \frac{1}{\lambda t}\nabla\ell(w)$.

To apply Pegasos in the partial information case we could simply replace the gradient vector $\nabla\ell(w)$ with its estimation given in (8). However, our analysis shows that it is desirable to maintain an estimation vector $\widetilde{\nabla}\ell(w)$ with small magnitude. Since the magnitude of $\widetilde{\nabla}\ell(w) = 2\big(\hat{y} - y\big)v$ is order of $d\,\|w\|_1$, we would like to ensure that $\|w\|_1$ is always smaller than some threshold B. We achieve this goal by adding an additional projection step at the end of each Pegasos's iteration. Formally, the update is performed in two steps as follows

$$w \leftarrow \left(1 - \frac{1}{t}\right)w - \frac{1}{\lambda t}2\big(\hat{y} - y\big)v \tag{10}$$

$$w \leftarrow \mathop{\operatorname{argmin}}_{u:\,\|u\|_1 \le B} \|u - w\|_2 \tag{11}$$

ALGORITHM: AER(S,k)
INPUT: Training set S of size m, local budget $k \geq 2$ (with k even)
PARAMETER: $\lambda > 0$
INITIALIZATION: $w = 0 \in \mathbb{R}^d$; $\bar{w} = w$; $t = 1$

for each $(x,y) \in S$
  $v = 0 \in \mathbb{R}^d$; $\hat{y} = 0$
  Choose C uniformly at random from all subsets of $[d]$ of size $\frac{k}{2}$
  for each $j \in C$
    $v_j = v_j + \frac{2}{k} d\, x_j$
  end
  for $r = 1, \ldots, k/2$
    sample i from $[d]$ based on $P(i) = \frac{|w_i|}{\|w\|_1}$ (if $w = 0$ set $P(i) = 1/d$)
    $\hat{y} = \hat{y} + \frac{2}{k} \operatorname{sgn}(w_i) \|w\|_1 x_i$
  end
  $w = \left(1 - \frac{1}{t}\right) w - \frac{2}{\lambda t}(\hat{y} - y) v$
  $w = \operatorname*{argmin}_{u:\|u\|_1 \leq B} \|u - w\|_2$
  $\bar{w} = \bar{w} + \frac{w}{m}$; $t = t + 1$
end

OUTPUT: $\bar{w}$

Figure 2: An adaptation of the Pegasos algorithm to the local budget setting. Theorem 12 provides a performance guarantee for this algorithm.

where v and $\hat{y}$ are respectively defined by (6) and (7). The projection step (11) can be performed efficiently in time $O(d)$ using the technique described in Duchi et al. (2008). A pseudo-code of the resulting **A**ttribute **E**fficient **R**egression algorithm is given in Figure 2.

Note that the right-hand side of (10) is $w - \frac{1}{\lambda t} \nabla f$ for the function

$$f(w) = \tfrac{\lambda}{2} \|w\|_2^2 + 2(\hat{y} - y)\langle v, w \rangle \ . \tag{12}$$

This observation is used in the proof of the following result, providing convergence guarantees for AER.

Theorem 12 *Let $\mathcal{D}$ be a distribution on pairs $(x,y) \in \mathbb{R}^d \times \mathbb{R}$ such that $\|x\|_\infty \leq 1$ and $|y| \leq B$ with probability one. Let S be a training set of size m and let $\bar{w}$ be the output of AER(S,k) run with $\lambda = 12d\sqrt{\log(m)/(mk)}$. Then, there exists a constant $c > 0$ such that*

$$L_{\mathcal{D}}(\bar{w}) \leq \min_{w:\|w\|_1 \leq B} L_D(w) + c\, d B^2 \sqrt{\frac{1}{km} \ln \frac{m}{\delta}}$$

holds with probability at least $1-\delta$ *over both the choice of the training set and the algorithm's own randomization.*

Proof Let $y_t, \hat{y}_t, v_t, w_t$ be the values of $y, \hat{y}, \underset{\sim}{y}, w$, respectively, at each iteration t of the AER algorithm. Moreover, let $\nabla_t = 2(\langle w_t, x_t\rangle - y_t)x_t$ and $\widetilde{\nabla}_t = 2(\hat{y}_t - y_t)v_t$. From the convexity of the squared loss, and taking expectation with respect to the algorithm's own randomization, we have that for any vector $w^\star$ such that $\|w^\star\|_1 \le B$,

$$\begin{aligned}\mathbb{E}\left[\sum_{t=1}^m \big(\langle w_t, x_t\rangle - y_t\big)^2\right] - \sum_{t=1}^m \big(\langle w^\star, x_t\rangle - y_t\big)^2 &\le \mathbb{E}\left[\sum_{t=1}^m \langle \nabla_t, w_t - w^\star\rangle\right] \\ &= \mathbb{E}\left[\sum_{t=1}^m \langle \widetilde{\nabla}_t, w_t - w^\star\rangle\right] \\ &= \mathbb{E}\left[\sum_{t=1}^m 2(\hat{y}_t - y_t)\langle v_t, w_t - w^\star\rangle\right] .\end{aligned}$$

For the first equality we used Lemma 11, which states that, conditioned on w_t, $\mathbb{E}\big[\widetilde{\nabla}_t\big] = \nabla_t$.

We now deterministically bound the random quantity inside the above expectation as follows

$$\begin{aligned}\sum_{t=1}^m 2(\hat{y}_t - y_t)\langle v_t, w_t - w^\star\rangle &= \sum_{t=1}^m \left(\frac{\lambda}{2}\|w_t\|_2^2 + 2(\hat{y}_t - y_t)\langle v_t, w_t\rangle\right) \\ &\quad - \sum_{t=1}^m \left(\frac{\lambda}{2}\|w^\star\|_2^2 + 2(\hat{y}_t - y_t)\langle v_t, w^\star\rangle\right) + m\frac{\lambda}{2}\|w^\star\|_2^2 \\ &= \sum_{t=1}^m f_t(w_t) - \sum_{t=1}^m f_t(w^\star) + m\frac{\lambda}{2}\|w^\star\|_2^2\end{aligned}$$

where $f_t(w) = \frac{\lambda}{2}\|w\|_2^2 + 2(\hat{y}_t - y_t)\langle v_t, w\rangle$ is the λ-strongly convex function defined in (12). Recalling that the right-hand side in the AER update (10) is equal to $w_t - \frac{1}{\lambda t}\nabla f_t(w_t)$, we can apply the following logarithmic regret bound for λ-strongly convex functions (Hazan et al., 2006; Kakade and Shalev-Shwartz, 2008)

$$\sum_{t=1}^m f_t(w_t) - \sum_{t=1}^m f_t(w^\star) \le \frac{1}{\lambda}\left(\max_t \|\nabla f_t(w_t)\|^2\right)\ln m$$

which remains valid also in the presence of the projection steps (11). Similarly to the analysis of Pegasos, and using our assumptions on $\|x_t\|_\infty$ and $|y_t|$, the norm of the gradient $\nabla f_t(w_t)$ is bounded as follows

$$\big\|\nabla f_t(w_t)\big\| \le \lambda\|w_t\| + 2\big|\hat{y}_t - y_t\big|\,\|v_t\| \le \lambda\|w_t\| + 4Bd\sqrt{\frac{2}{k}} .$$

In addition, it is easy to verify (e.g., using an iductive argument) that

$$\|w_t\| \le \frac{1}{\lambda}4Bd\sqrt{\frac{2}{k}},$$

which yields

$$\big\|\nabla f_t(w_t)\big\| \le 8Bd\sqrt{\frac{2}{k}} .$$

This gives the bound

$$\sum_{t=1}^{m} 2(\hat{y}_t - y_t)\langle v_t, w_t - w^\star\rangle \leq \frac{128(dB)^2}{\lambda k}\ln m + m\frac{\lambda}{2}\|w^\star\|_2^2 .$$

Choosing $\lambda = 16d\sqrt{\log(m)/(km)}$ and noting that $\|\cdot\|_2 \leq \|\cdot\|_1$ we get that

$$\sum_{t=1}^{m} 2(\hat{y}_t - y_t)\langle v_t, w_t - w^\star\rangle \leq 16dB^2\sqrt{\frac{m}{k}\ln m} .$$

The resulting bound is then

$$\mathbb{E}\left[\sum_{t=1}^{m}\left(\langle w_t, x_t\rangle - y_t\right)^2\right] \leq \sum_{t=1}^{m}\left(\langle w^\star, x_t\rangle - y_t\right)^2 + 16dB^2\sqrt{\frac{m}{k}\ln m} .$$

To conclude the proof, we apply the online-to-batch conversion of Cesa-Bianchi et al. (2004, Corollary 2) to the probability space that includes both the algorithm's own randomization and the product distribution from which the training set is drawn. Since $\left(\langle w, x_t\rangle - y_t\right)^2 \leq 4B^2$ for all w such that $\|w\|_1 \leq B$ (recall our assumptions on x_t and y_t), and using the convexity of the square loss, we obtain that

$$L_{\mathcal{D}}(\bar{w}) \leq \inf_{w:\|w\|\leq B} L_{\mathcal{D}}(w) + 16dB^2\sqrt{\frac{1}{km}\ln m} + 4B^2\sqrt{\frac{2}{m}\ln\frac{1}{\delta}}$$

holds with probability at least $1-\delta$ with respect to all random events. ■

Note that for small values of k (which is the reasonable regime here) the bound for AER is much better than the bound for Baseline: ignoring logarithmic factors, instead of quadratic dependence on d, we have only linear dependence on d.

It is interesting to compare the bound for AER to the Lasso bound (2) for the full information case. As it can be seen, to achieve the same level of risk, AER needs a factor of d^2/k more examples than the full information Lasso.[4] Since each AER example uses only k attributes while each Lasso example uses all d attributes, the ratio between the total number of *attributes* AER needs and the number of attributes Lasso needs to achieve the same error is $O(d)$. Intuitively, when having d times total number of attributes, we can fully compensate for the partial information protocol.

However, in some situations even this extra d factor is not needed. Indeed, suppose we know that the vector $w^\star$, which minimizes the risk, is dense. That is, it satisfies $\|w^\star\|_1 \approx \sqrt{d}\,\|w^\star\|_2 \leq B$. In this case, by setting $\lambda = d^{3/2}\sqrt{\log(m)/(km)}$, and using the tighter bound $\|w^\star\|_2 \leq B/\sqrt{d}$ instead of $\|w^\star\|_2 \leq \|w^\star\|_1 \leq B$ in the proof of Theorem 12, we get a final bound of the form

$$L_{\mathcal{D}}(\bar{w}) \leq L_{\mathcal{D}}(w^\star) + cB^2\sqrt{\frac{d}{km}\ln\frac{m}{\delta}} .$$

Therefore, the number of examples AER needs in order to achieve the same error as Lasso is only a factor d/k more than the number of examples Lasso uses. But, this implies that both AER and Lasso needs the same number of *attributes* in order to achieve the same level of error! Crucially, the above holds only if $w^\star$ is dense. When $w^\star$ is sparse we have $\|w^\star\|_1 \approx \|w^\star\|_2$ and then AER needs more attributes than Lasso.

4. We note that when $d = k$ we still do not recover the full information bound. However, it is possible to improve the analysis and replace the factor $d/\sqrt{k}$ with a factor $d\left(\max_t \|x_t\|_2\right)/k$.

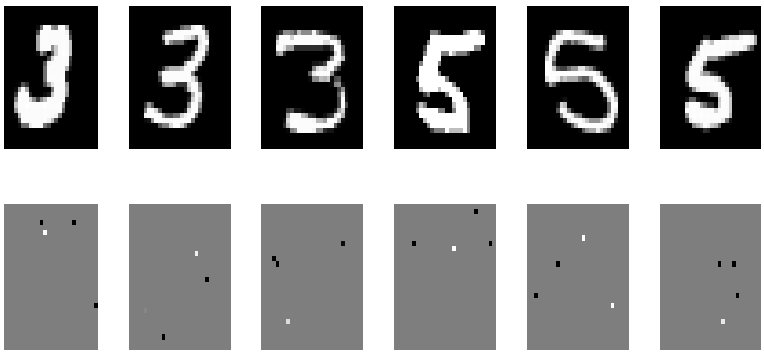

Figure 3: In the upper row six examples from the training set (of digits 3 and 5) are shown. In the lower row we show the same six examples, where only four randomly sampled pixels from each original image are displayed.

7. Experiments

We performed some experiments to test the behavior of our algorithm on the well-known MNIST digit recognition data set (Le Cun et al., 1998), which contains 70,000 images (28×28 pixels each) of the digits $0-9$. The advantages of this data set for our purposes is that it is not a small scale data set, has a reasonable dimensionality-to-data-size ratio, and the setting is clearly interpretable graphically. While this data set is designed for classification (e.g., recognizing the digit in the image), we can still apply our algorithms on it by regressing to the label.

First, to demonstrate the hardness of our settings, we provide in Figure 3 below some examples of images from the data set, in the full information setting and the partial information setting. The upper row contains six images from the data set, as available to a full information algorithm. A partial information algorithm, however, will have a much more limited access to these images. In particular, if the algorithm may only choose $k = 4$ pixels from each image, the same six images as available to it might look like the bottom row of Figure 3.

We began by looking at a data set composed of "3" vs. "5", where all the "3" digits were labeled as -1 and all the "5" digits were labeled as $+1$. We ran four different algorithms on this data set: the simple Baseline algorithm, AER, as well as ridge regression and Lasso for comparison (for Lasso, we solved (1) with $p = 1$). Both ridge regression and Lasso were run in the full information setting: Namely, they enjoyed full access to all attributes of all examples in the training set. The Baseline algorithm and AER, however, were given access to only four attributes from each training example.

We randomly split the data set into a training set and a test set (with the test set being 10% of the original data set). For each algorithm, parameter tuning was performed using 10-fold cross validation. Then, we ran the algorithm on increasingly long prefixes of the training set, and measured the average regression error $(\langle w,x\rangle - y)^2$ on the test set. The results (averaged over runs on 10 random train-test splits) are presented in Figure 4. In the upper plot, we see how the test regression error improves with the number of examples. The Baseline algorithm is highly unstable at the beginning, probably due to the ill-conditioning of the estimated covariance matrix, although it eventually stabilizes (to prevent a graphical mess at the left hand side of the figure, we removed the error bars from the corresponding plot). Its performance is worse than AER, completely in line with our earlier theoretical analysis.

The bottom plot of Figure 4 is similar, only that now the X-axis represents the accumulative number of attributes seen by each algorithm rather than the number of examples. For the partial-information algorithm, the graph ends at approximately 49,000 attributes, which is the total number of attributes accessed by the algorithm after running over all training examples, seeing $k = 4$ pixels from each example. However, for the full-information algorithms 49,000 attributes are already seen after just 62 examples. When we compare the algorithms in this way, we see that our AER algorithm achieves excellent performance for a given attribute budget, significantly better than the other 1-norm-based algorithms (Baseline and Lasso). Moreover, AER is even comparable to the full information 2-norm-based ridge regression algorithm, which performs best on this data set.

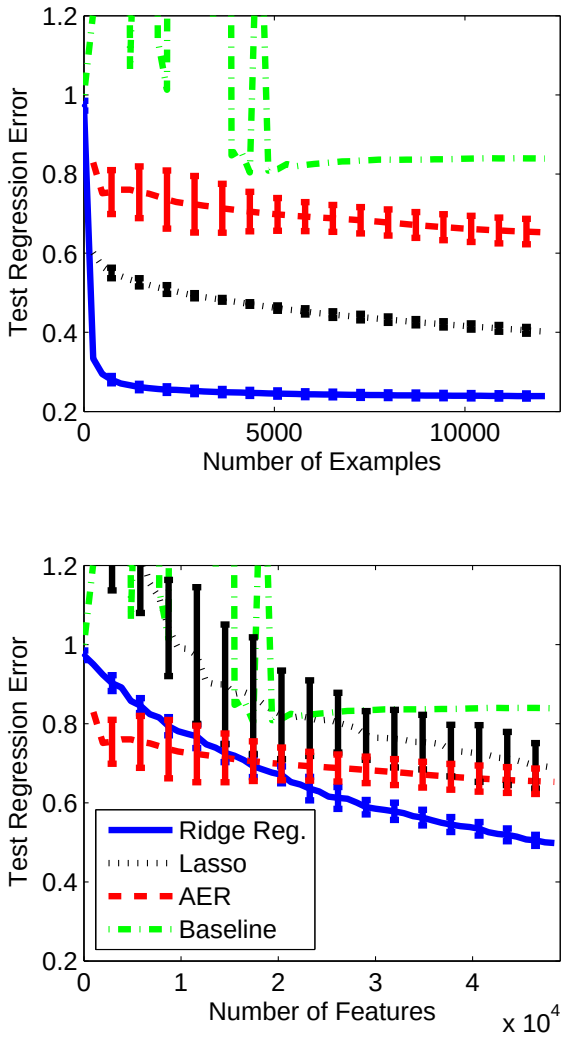

Figure 4: Test regression error for each one of the four algorithms (ridge regression, Lasso, AER, and Baseline), over increasing prefixes of the training set for "3" vs. "5". The results are averaged over 10 runs.

Finally, we tested the algorithms over 45 data sets generated from MNIST, one for each possible pair of digits. For each data set and each of 10 random train-test splits, we performed parameter tuning for each algorithm separately, and checked the average squared error on the test set. The median test errors over all data sets are presented in the table below.

		Test Error
Full Information	**Ridge**	0.110
	Lasso	0.222
Partial Information	**AER**	0.320
	Baseline	0.812

As can be seen, the AER algorithm manages to achieve good performance, not much worse than the full information Lasso algorithm. The Baseline algorithm, however, achieves a substantially worse performance, in line with our theoretical analysis above. We also calculated the test classification error of AER, that is, $\text{sign}(\langle w,x\rangle) \neq y$, and found out that AER, which can see only 4 pixels per image, usually performs only a little worse than the full information algorithms (ridge regression and Lasso), which enjoy full access to all 784 pixels in each image. In particular, the median test classification errors of AER, Lasso, and Ridge are 3.5%, 1.1%, and 1.3% respectively.

8. Discussion and Extensions

In this paper we have investigated three budgeted learning settings with different constraints on the way instance attributes may be accessed: a local constraint on each training example (local budget), a global constraint on the set of all training examples (global budget), and a constraint on each test example (prediction on a budget). In the local budget setting, we have introduced a simple and efficient algorithm, AER, that learns by accessing a pre-specified number of attributes from each training example. The AER algorithm comes with formal guarantees, is provably competitive with algorithms which enjoy full access to the data, and performs well in simple experiments. This result is complemented by a general lower bound for the global budget setting which is a factor d smaller than the upper bound achieved by our algorithm. We note that this gap has been recently closed by Hazan and Koren (2011), which in our local budget setting, show 1-norm and 2-norm-based algorithms for learning linear predictors using only $\widetilde{O}(d)$ attributes, thus matching our lower bound to within logarithmic factors.

Whereas AER is based on Pegasos, our adaptive sampling approach easily extends to other gradient-based algorithms. For example, generalized additive algorithms such as p-norm Perceptrons and Winnow—see, for example, Cesa-Bianchi and Lugosi (2006).

In contrast to the local/global budget settings, where we can learn efficiently by accessing few attributes of each training example, we showed that accessing a limited number of attributes at test time is a significantly harder setting. Indeed, we proved that is not possible to build an active linear predictor that uses two attributes of each test example and whose error is smaller than a certain constant, even when there exists a linear predictor achieving zero error on the same data source.

An obvious direction for future research is how to deal with loss functions other than the squared loss. In related work (Cesa-Bianchi et al., 2010), we developed a technique which allows us to deal with arbitrary analytic loss functions. However, in the setting of this paper, those techniques would lead to sample complexity bounds which are exponential in d. Another interesting extension we are considering is connecting our results to the field of privacy-preserving learning (Dwork,

2008), where the goal is to exploit the attribute efficiency property in order to prevent acquisition of information about individual data instances.

Acknowledgments

We would like to thank the anonymous reviewers for their detailed and helpful comments. N. Cesa-Bianchi gratefully acknowledges partial support by the PASCAL2 Network of Excellence under EC grant no. 216886. This publication only reflects the authors' views.

Appendix A. Proof of Theorem 1

The outline of the proof is as follows. We define a specific distribution such that only one "good" feature is slightly correlated with the label. We then show that if some algorithm learns a linear predictor with an extra risk of at most ε, then it must know the value of the good feature. Next, we construct a variant of a multi-armed bandit problem out of our distribution and show that a good learner can yield a good prediction strategy. Finally, we adapt a lower bound for the multi-armed bandit problem given in Auer et al. (2003), to conclude that the number k of attributes viewed by a good learner must satisfy $k = \Omega\left(\frac{d}{\varepsilon}\right)$.

A.1 The Distribution

We generate a joint distribution over $\mathbb{R}^d \times \mathbb{R}$ as follows. Choose some $j \in [d]$. First, we generate $y_1, y_2, \ldots \in \{\pm 1\}$ i.i.d. according to $\mathbb{P}[y_t = 1] = \mathbb{P}[y_t = -1] = \frac{1}{2}$. Given j and y_t, $x_t \in \{\pm 1\}$ is generated according to $\mathbb{P}[x_{t,i} = y_t] = \frac{1}{2} + \mathbb{1}\{i = j\}p$ where $p > 0$ is chosen later. Denote by $\mathbb{P}_j$ the distribution mentioned above assuming the "good" feature is j. Also denote by $\mathbb{P}_u$ the uniform distribution over $\{\pm 1\}^{d+1}$. Analogously, we denote by $\mathbb{E}_j$ and $\mathbb{E}_u$ expectations w.r.t. $\mathbb{P}_j$ and $\mathbb{P}_u$.

A.2 A Good Regressor "Knows" j

We now show that if we have a good linear regressor than we can know the value of j. It is easy to see that the optimal linear predictor under the distribution $\mathbb{P}_j$ is $w^\star = 2pe^j$, and the risk of $w^\star$ is

$$L_{\mathbb{P}_j}(w^\star) = \mathbb{E}_j\left[(\langle w^\star, x\rangle - y)^2\right] = \left(\tfrac{1}{2}+p\right)(1-2p)^2 + \left(\tfrac{1}{2}-p\right)(1+2p)^2 = 1+4p^2-8p^2 = 1-4p^2 \,.$$

The risk of an arbitrary weight vector w under P_j is

$$L_{\mathbb{P}_j}(w) = \mathbb{E}_j\left[(\langle w, x\rangle - y)\right]^2 = \sum_{i\neq j} w_i^2 + \mathbb{E}_j\left[(w_j x_j - y)^2\right] = \sum_{i\neq j} w_i^2 + w_j^2 + 1 - 4pw_j \,.$$

Suppose that $L_{\mathbb{P}_j}(w) - L_{\mathbb{P}_j}(w^\star) < \varepsilon$. This implies that:

1. For all $i \neq j$ we have $w_i^2 < \varepsilon$, or equivalently, $|w_i| < \sqrt{\varepsilon}$.
2. $1 + w_j^2 - 4pw_j - (1-4p^2) < \varepsilon$ and thus $|w_j - 2p| < \sqrt{\varepsilon}$ which gives $|w_j| > 2p - \sqrt{\varepsilon}$.

By choosing $p = \sqrt{\varepsilon}$, the above implies that we can identify the value of j from any w whose risk is strictly smaller than $L_{\mathbb{P}_j}(w^\star) + \varepsilon$.

A.3 Constructing A Variant Of A Multi-Armed Bandit Problem

We now construct a variant of the multi-armed bandit problem out of the distribution $\mathbb{P}_j$. Each coordinate $i \in \{1,\ldots,d\}$ is an arm and the reward of pulling i at time t is $\mathbb{1}\{x_{N_{i,t},i} = y_{N_{i,t}}\} \in \{0,1\}$, where $N_{i,t}$ denotes the random number of times arm i has been pulled in the first t plays. Hence the expected reward of pulling i is $\frac{1}{2} + \mathbb{1}\{i = j\}p$. At the end of each round t the player observes $x_{N_{i,t},i}$ and $y_{N_{i,t}}$.

A.4 A Good Learner Yields A Bandit Strategy

Suppose that we have a learner that, for any $j = 1,\ldots,d$, can learn a linear predictor with $L_{\mathbb{P}_j}(w) - L_{\mathbb{P}_j}(w^\star) < \varepsilon$ using k attributes. Since we have shown that once $L_{\mathbb{P}_j}(w) - L_{\mathbb{P}_j}(w^\star) < \varepsilon$ we know the value of j, we can construct a strategy for the multi-armed bandit problem in a straightforward way. Simply use the first m examples to learn w and from then on always pull the arm j. The expected reward of this strategy under any $\mathbb{P}_j$ after $T \geq k$ plays is at least

$$\frac{k}{2} + (T-k)\left(\frac{1}{2} + p\right) = \frac{T}{2} + (T-k)p\,. \tag{13}$$

A.5 An Upper Bound On the Reward Of Any Bandit Strategy

Recall that under distribution $\mathbb{P}_j$ the expected reward for pulling arm I is $\frac{1}{2} + p\mathbb{1}\{I = j\}$. Hence, the total expected reward of a player that runs for T rounds is upper bounded by $\frac{1}{2}T + p\mathbb{E}_j[N_j]$, where $N_j = N_{j,T}$ is the overall number of pulls of arm j. Moreover, at the end of each round t the player observes $x_{s,i}$ and y_s, where $s = N_{i,t}$. This allows the player to compute the value of the reward for the current play. For any s, note that y_s is observed whenever some arm i is pulled for the s-th time. However, since $\mathbb{P}_j\left[x_{i,s} = y_s\right] = \mathbb{P}_j\left[x_{i,s} = y_s \mid y_s\right]$ for all i (including $i = j$), the knowledge of y_s does not provide any information about the distribution of rewards for arm i. Therefore, without loss of generality, we can assume that at each play the bandit strategy observes only the obtained binary reward. This implies that our bandit construction is identical to the one used in the proof of Theorem 5.1 in Auer et al. (2003). In particular, for any bandit strategy there exists some arm j such that the expected reward of the strategy under distribution $\mathbb{P}_j$ is at most

$$\frac{T}{2} + p\left(\frac{T}{d} + T\sqrt{-\frac{T}{d}\ln(1-4p^2)}\right) \leq \frac{T}{2} + p\left(\frac{T}{d} + T\sqrt{\frac{6T}{d}p^2}\right) \tag{14}$$

where we used the inequality $-\ln(1-q) \leq \frac{3}{2}q$ for $q \in [0,1/4]$. Note that $q = 4p^2 = 4\varepsilon \in [0,1/4]$ when $\varepsilon \leq 1/16$.

A.6 Concluding The Proof

Take a learning algorithm that finds an ε-good predictor using k attributes. Since the reward of the strategy based on this learning algorithm cannot exceed the upper bound given in (14), from (13) we obtain that

$$\frac{T}{2} + (T-k)p \leq \frac{T}{2} + p\left(\frac{T}{d} + T\sqrt{\frac{6T}{d}p^2}\right)$$

which solved for k gives

$$k \geq T\left(1 - \frac{1}{d} - \sqrt{\frac{6T}{d}p^2}\right) .$$

Since we assume $d \geq 4$, choosing $T = \lfloor d/(96p^2) \rfloor$, and recalling $p^2 = \varepsilon$, gives

$$k \geq \frac{T}{2} = \frac{1}{2}\left\lfloor \frac{d}{96\varepsilon} \right\rfloor .$$

References

P. Auer, N. Cesa-Bianchi, Y. Freund, and R.E. Schapire. The nonstochastic multiarmed bandit problem. *SIAM Journal on Computing*, 32, 2003.

M-F Balcan, A. Beygelzimer, and J. Langford. Agnostic active learning. In *ICML*, 2006.

S. Ben-David and E. Dichterman. Learning with restricted focus of attention. *JCSS: Journal of Computer and System Sciences*, 56, 1998.

A. Beygelzimer, S. Dasgupta, and J. Langford. Importance weighted active learning. In *ICML*, 2009.

R. Calderbank, S. Jafarpour, and R. Schapire. Compressed learning: Universal sparse dimensionality reduction and learning in the measurement domain. Manuscript, 2009.

N. Cesa-Bianchi and G. Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, 2006.

N. Cesa-Bianchi, A. Conconi, and C. Gentile. On the generalization ability of on-line learning algorithms. *IEEE Transactions on Information Theory*, 50(9):2050–2057, September 2004.

N. Cesa-Bianchi, S. Shalev-Shwartz, and O. Shamir. Online learning of noisy data with kernels. In *COLT*, 2010.

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

K. Deng, C. Bourke, S. Scott, J. Sunderman, and Y. Zheng. Bandit-based algorithms for budgeted learning. In *ICDM*, 2007.

L. Devroye, L. Györfi, and G. Lugosi. *A Probabilistic Theory of Pattern Recognition*. Springer, 1996.

J. Duchi, S. Shalev-Shwartz, Y. Singer, and T. Chandra. Efficient projections onto the ℓ_1-ball for learning in high dimensions. In *ICML*, 2008.

C. Dwork. Differential privacy: A survey of results. In M. Agrawal, D.-Z. Du, Z. Duan, and A. Li, editors, *TAMC*, volume 4978 of *Lecture Notes in Computer Science*, pages 1–19. Springer, 2008.

R. Greiner, A. J. Grove, and D. Roth. Learning cost-sensitive active classifiers. *Artificial Intelligence*, 139(2):137–174, 2002.

S. Guha and K. Munagala. Approximation algorithms for budgeted learning problems. In *STOC*, 2007.

S. Hanneke. A bound on the label complexity of agnostic active learning. In *ICML*, 2007.

S. Hanneke. Adaptive rates of convergence in active learning. In *COLT*, 2009.

D. Haussler. Decision theoretic generalizations of the PAC model for neural net and other learning applications. *Information and Computation*, 100(1):78–150, 1992.

E. Hazan and T. Koren. Optimal algorithms for ridge and Lasso regression with partially observed attributes. *CoRR*, abs/1108.4559, 2011.

E. Hazan, A. Kalai, S. Kale, and A. Agarwal. Logarithmic regret algorithms for online convex optimization. In *COLT*, 2006.

S.M. Kakade and S. Shalev-Shwartz. Mind the duality gap: Logarithmic regret algorithms for online optimization. In *Advances in Neural Information Processing Systems 22*, 2008.

S.M. Kakade, K. Sridharan, and A. Tewari. On the complexity of linear prediction: Risk bounds, margin bounds, and regularization. In *NIPS*, 2008.

A. Kapoor and R. Greiner. Learning and classifying under hard budgets. In *ECML*, 2005a.

A. Kapoor and R. Greiner. Budgeted learning of bounded active classifiers. In *Proceedings of the ACM SIGKDD Workshop on Utility-Based Data Mining*, 2005b.

Y. L. Le Cun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of IEEE*, 86(11):2278–2324, November 1998.

O. Madani, D.J. Lizotte, and R. Greiner. Active model selection. In *UAI*, 2004.

S. Shalev-Shwartz, Y. Singer, and N. Srebro. Pegasos: Primal Estimated sub-GrAdient SOlver for SVM. In *ICML*, 2007.

R. Tibshirani. Regression shrinkage and selection via the lasso. *J. Royal. Statist. Soc B.*, 58(1): 267–288, 1996.

V. N. Vapnik. *Statistical Learning Theory*. Wiley, 1998.

S. Zhou, J. Lafferty, and L. Wasserman. Compressed and privacy-sensitive sparse regression. *IEEE Transactions on Information Theory*, 55(2):846–866, 2009.