

# Quantity Makes Quality: Learning with Partial Views

**Nicolò Cesa-Bianchi**

DSI  
Università degli Studi di Milano, Italy

**Shai Shalev-Shwartz**

School of Computer Science & Engineering  
The Hebrew University, Israel

**Ohad Shamir**

Microsoft Research New-England  
USA

## Abstract

In many real world applications, the number of examples to learn from is plentiful, but we can only obtain limited information on each individual example. We study the possibilities of efficient, provably correct, large-scale learning in such settings. The main theme we would like to establish is that large amounts of examples can compensate for the lack of full information on each individual example. The type of partial information we consider can be due to inherent noise or from constraints on the type of interaction with the data source. In particular, we describe and analyze algorithms for budgeted learning, in which the learner can only view a few attributes of each training example (Cesa-Bianchi, Shalev-Shwartz, and Shamir 2010a; 2010c), and algorithms for learning kernel-based predictors, when individual examples are corrupted by random noise (Cesa-Bianchi, Shalev-Shwartz, and Shamir 2010b).

## Introduction

In recent years, large amounts of data are collected from various sources and information overload is quickly exacerbating. Machine learning can and should play a central role in analyzing this data and in performing useful inferences based on it. However, in many cases we do not have full information on each individual example, and most traditional learning methods are incapable of managing efficiently data sets with partial information. The ability to manage large amounts of data and furthermore, to harness the amounts of data into more efficient algorithms is key for making progress in machine learning at the data revolution era.

It is a well-known fact that training on more data improves the accuracy of learning algorithms. In this paper we demonstrate how one can use more data to compensate for lack of full information on each individual example.

Many methods have been proposed for dealing with missing or partial information. Most of the approaches do not come with formal guarantees on the risk of the resulting algorithm and are not guaranteed to converge in a polynomial time. The difficulty stems from the exponential number of ways to complete the missing information. In the framework of generative models, a popular approach

is the Expectation-Maximization (EM) procedure (Dempster, Laird, and Rubin 1977). The main drawback of the EM approach is that it might find sub-optimal solutions. Even elegant methods that have some convergence guarantees, such as graph based approaches, usually do not scale well with the size of the dataset —see e.g. (Zhu 2006; Delalleau, Bengio, and Roux 2005; Zhu and Lafferty 2005). In contrast, we will propose novel discriminative methods for dealing with missing information that come with both statistical and computational convergence guarantees. Furthermore, the main theme we would like to establish is that even for the purpose of improving computational efficiency, large amounts of data should be an asset rather than a burden. We shall present cases in which the availability of many examples compensates for the lack of full information on each individual example.

Roughly speaking, we use the following techniques for harnessing the availability of more data:

- **Missing information as noise:** The lack of full information on each individual example can stem from various reasons. In some situations, the instances are corrupted by noise. In other situations, the lack of information is due to constraints on the type of interaction with the data source. As we will show later, even in such cases, we are sometimes able to model the missing information as an additive noise. Viewing the missing information as noise, it is clear that more data can compensate for the noise by reducing the variance in the estimate of the quantities of interest.
- **Active acquisition of information:** In many applications we can actively control what partial information we get for each example. Roughly speaking, given a set of possible partial views of an example, we will actively pick a view in a randomized way, so as to construct a noisy estimate of the *entire information* contained in the example. Put another way, instead of learning using the original parametrization in which some information is missing, we construct a different parametrization in which the missing information is replaced by noise. Therefore, instead of receiving the exact value of each individual example in a small set of examples it suffices to get noisy estimates of the values of a large number of examples. Technically, we borrow and generalize ideas from the adversarial multi-

armed bandit setting (Auer et al. 2003).

In the next sections we give rigorous meaning to the aforementioned intuitive ideas by analyzing two learning problems with partial information.

### Attribute Efficient Learning

Suppose we would like to predict if a person has some disease based on medical tests. Theoretically, we may choose a sample of the population, perform a large number of medical tests on each person in the sample and learn from this information. In many situations this is unrealistic, since patients participating in the experiment are not willing to go through a large number of medical tests. The above example motivates the learning with partial information scenario in which there is a hard constraint on the number of attributes the learner may view for each training example.

We propose an efficient algorithm for linear regression, dealing with this partial information problem, and bound the number of additional training examples sufficient to compensate for the lack of full information on each training example. As we said in the introduction, we actively pick which attributes to observe in a randomized way so as to construct a “noisy” version of *all* attributes. We can still learn despite the error of this estimate because we use a larger set of noisy training examples.

We start with a formal description of the learning problem. In linear regression each example is an instance-target pair,  $(\mathbf{x}, y) \in \mathbb{R}^d \times \mathbb{R}$ . We refer to  $\mathbf{x}$  as a vector of attributes and the goal of the learner is to find a linear predictor  $\mathbf{x} \mapsto \langle \mathbf{w}, \mathbf{x} \rangle$ , where we refer to  $\mathbf{w} \in \mathbb{R}^d$  as the predictor. The performance of a predictor  $\mathbf{w}$  on an instance-target pair,  $(\mathbf{x}, y) \in \mathbb{R}^d \times \mathbb{R}$ , is measured by the square loss  $(\langle \mathbf{w}, \mathbf{x} \rangle - y)^2$ . Following the standard framework of statistical learning (Haussler 1992; Devroye, Györfi, and Lugosi 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:  $L_{\mathcal{D}}(\mathbf{w}) \stackrel{\text{def}}{=} \mathbb{E}_{(\mathbf{x}, y) \sim \mathcal{D}}[(\langle \mathbf{w}, \mathbf{x} \rangle - y)^2]$ . Since the distribution  $\mathcal{D}$  is unknown to the learner he learns by relying on a training set of  $m$  examples  $S = (\mathbf{x}_1, y_1), \dots, (\mathbf{x}_m, y_m)$ , which are assumed to be sampled i.i.d. from  $\mathcal{D}$ . We now distinguish between two scenarios:

- **Full information:** The learner receives the entire training set. This is the traditional linear regression setting.
- **Partial information:** For each individual example,  $(\mathbf{x}_i, y_i)$ , the learner receives the target  $y_i$  but is only allowed to see  $k$  attributes of  $\mathbf{x}_i$ , where  $k$  is a parameter of the problem. The learner has the freedom to actively choose *which* of the attributes will be revealed, as long as at most  $k$  of them will be given.

A popular approach for learning in the full information case is the so-called Empirical Risk Minimization (ERM) rule, in which one minimizes the empirical loss on the training set over a predefined hypothesis class. For example, the Lasso algorithm can be shown to be equivalent to ERM with the hypothesis class being  $\{\mathbf{w} : \|\mathbf{w}\|_1 \leq B\}$ , for some parameter  $B$ . Standard risk bounds for Lasso imply that if  $\hat{\mathbf{w}}$  is

its solution, then with probability greater than  $1 - \delta$  over the choice of a training set of size  $m$  we have

$$L_{\mathcal{D}}(\hat{\mathbf{w}}) \leq \min_{\mathbf{w}: \|\mathbf{w}\|_1 \leq B} L_{\mathcal{D}}(\mathbf{w}) + O\left(B^2 \sqrt{\frac{\ln(d/\delta)}{m}}\right). \quad (1)$$

In the partial information case, we follow a different approach. Recall that our goal is to minimize the true risk  $L_{\mathcal{D}}(\mathbf{w})$ . Had we known the distribution  $\mathcal{D}$  we could have minimized  $L_{\mathcal{D}}$ , e.g., using gradient descent. Since we do not know  $\mathcal{D}$ , but can only get samples from it, we adapt a stochastic gradient descent (SGD) approach. SGD is an iterative algorithm where at each step we update the current solution based on an unbiased estimate of the gradient of the objective function.

The gradient of  $L_{\mathcal{D}}(\mathbf{w})$  is the vector  $\mathbb{E}_{(\mathbf{x}, y) \sim \mathcal{D}}[(\langle \mathbf{w}, \mathbf{x} \rangle - y)\mathbf{x}]$ . We now show how to construct an unbiased estimate of this vector. For simplicity we focus on the case in which we can view two features per each example. We first sample a random example  $(\mathbf{x}, y) \sim \mathcal{D}$ . Then, we estimate the vector  $\mathbf{x}$  while viewing a single attribute of it as follows. Pick an index  $i$  uniformly at random from  $[d] = \{1, \dots, d\}$  and define the estimation to be a vector  $\mathbf{v}$  such that

$$v_r = \begin{cases} dx_r & \text{if } r = i \\ 0 & \text{else} \end{cases}. \quad (2)$$

It is easy to verify that  $\mathbf{v}$  is an unbiased estimate of  $\mathbf{x}$ , namely,  $\mathbb{E}[\mathbf{v}] = \mathbf{x}$  where expectation is with respect to the choice of the index  $i$ . When we are allowed to see  $k > 1$  attributes, we simply repeat the above process (without replacement) and set  $\mathbf{v}$  to be the averaged vector.

Next, we show how to estimate the scalar  $(\langle \mathbf{w}, \mathbf{x} \rangle - y)$  while again viewing a single attribute of  $\mathbf{x}$ . Each vector  $\mathbf{w}$  can define a probability distribution over  $[d]$  by letting  $\mathbb{P}[i] = |w_i|/\|\mathbf{w}\|_1$ . Pick  $j$  from  $[d]$  according to the distribution defined by  $\mathbf{w}$ . Using  $j$  we estimate the term  $\langle \mathbf{w}, \mathbf{x} \rangle$  by  $\text{sgn}(w_j)\|\mathbf{w}\|_1 x_j$ . It is easy to verify that the expectation of the estimate equals  $\langle \mathbf{w}, \mathbf{x} \rangle$ . Overall, we have shown that if  $(\mathbf{x}, y)$  are chosen randomly from  $\mathcal{D}$ ,  $j$  is chosen uniformly at random from  $[d]$ , and  $i$  is chosen according to the distribution defined by  $\mathbf{w}$ , then the vector  $2(\text{sgn}(w_j)\|\mathbf{w}\|_1 x_j - y)\mathbf{v}$  is an unbiased estimate of the gradient,  $\nabla L_{\mathcal{D}}(\mathbf{w})$ .

Combining the above with standard convergence guarantees for SGD we obtain that the risk of the learnt vector satisfies:

$$L_{\mathcal{D}}(\bar{\mathbf{w}}) \leq \min_{\mathbf{w}: \|\mathbf{w}\|_1 \leq B} L_{\mathcal{D}}(\mathbf{w}) + O\left(\frac{dB^2}{\sqrt{k}} \sqrt{\frac{\ln(m/\delta)}{m}}\right)$$

with probability greater than  $1 - \delta$  over the choice of a training set of size  $m$ .

It is interesting to compare the bound for our algorithm to the Lasso bound in the full information case given in (1). As it can be seen, to achieve the same level of risk, the partial information algorithm needs a factor of  $d^2/k$  more examples than the full information Lasso. This quantifies the intuition that a larger number of examples can compensate for the lack of full information on each individual example.

Below, we present some of the experimental results obtained on the MNIST digit recognition dataset (Cun et

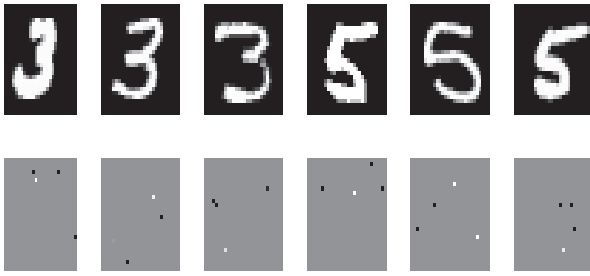


Figure 1: Full Information vs. Partial Information Setting.

al. 1998) (see (Cesa-Bianchi, Shalev-Shwartz, and Shamir 2010a) for a fuller experimental study). To demonstrate the hardness of our setting, we provide in Figure 1 below some examples of images from the dataset, in the full information setting and the partial information setting. The upper row contains six images from the dataset, 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 1.

Figure 2 exemplifies the type of results we obtain, on the dataset 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 dataset: a simple baseline algorithm which predicts based on the empirical covariance matrix (see (Cesa-Bianchi, Shalev-Shwartz, and Shamir 2010a) for full details), our algorithm discussed above (denoted as AER), as well as ridge regression and Lasso for comparison. 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 4 attributes from each training example. The  $X$ -axis represents the cumulative number of *attributes* seen by each algorithm. 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  $L_1$ -based algorithms, and even comparable to full-information ridge regression. However, while the full-information algorithms require all the information about each individual example, the AER algorithm can deal with partial information in each example.

### Kernel-Based Learning from Noisy Data

In many machine learning applications training data are typically collected by measuring certain physical quantities. Examples include bioinformatics, medical tests, robotics, and remote sensing. These measurements have errors that may be due to several reasons: sensor costs, communication constraints, privacy issues, or intrinsic physical limitations. In all such cases, the learner trains on a distorted version of the actual “target” data, which is where the learner’s predictive ability is eventually evaluated. In this work we investigate the extent to which a learning algorithm can achieve a good predictive performance when training data are cor-

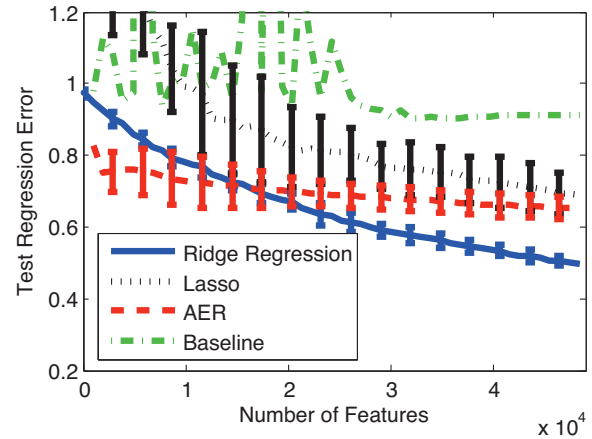


Figure 2: Test regression error over increasing prefixes of the training set for “3 vs. 5”. The results are averaged over 10 runs.

rupted by noise with unknown distribution.

We consider the problem of learning kernel-based predictors (Cristianini and Shawe-Taylor 2004; Schölkopf and Smola 2002). The kernel trick has had tremendous impact on machine learning theory and algorithms over the past decade. Unlike the standard setting, each time we sample a training example  $(\mathbf{x}, y)$  we are given access to  $y$  and to unbiased noisy measurements of  $\mathbf{x}$ . In a recent work (Cesa-Bianchi, Shalev-Shwartz, and Shamir 2010b) we have designed a general technique for learning kernel-based predictors from noisy data, where virtually nothing is known about the noise, except possibly an upper bound on its variance.

As in the attribute efficient learning problem described in the previous section, we rely on stochastic gradient estimates. However, due to the use of kernels, the construction of unbiased estimates here is much more involved. At the heart of these techniques lies an apparently little-known method from sequential estimation theory to construct unbiased estimates of non-linear and complex functions.

Suppose that we are given access to independent copies of a real random variable  $X$ , with expectation  $\mathbb{E}[X]$ , and some real function  $f$ , and we wish to construct an unbiased estimate of  $f(\mathbb{E}[X])$ . If  $f$  is a linear function, then this is easy: just sample  $x$  from  $X$ , and return  $f(x)$ . By linearity,  $\mathbb{E}[f(X)] = f(\mathbb{E}[X])$  and we are done. The problem becomes less trivial when  $f$  is a general, non-linear function, since usually  $\mathbb{E}[f(X)] \neq f(\mathbb{E}[X])$ . In fact, when  $X$  takes finitely many values and  $f$  is not a polynomial function, one can prove that no unbiased estimator can exist (see (Paninski 2003), Proposition 8 and its proof). Nevertheless, we show how in many cases one can construct an unbiased estimator of  $f(\mathbb{E}[X])$ , including cases covered by the impossibility result. There is no contradiction, because we do not construct a “standard” estimator. Usually, an estimator is a function from a given sample to the range of the parameter we wish to estimate. An implicit assumption is that the size of the sample given to it is fixed, and this is also a crucial ingredient in the impossibility result. We circumvent this by constructing an estimator based on a random number of samples.

Here is the key idea: suppose  $f : \mathbb{R} \rightarrow \mathbb{R}$  is any func-



tion continuous on a bounded interval. It is well known that one can construct a sequence of polynomials  $(Q_n(\cdot))_{n=1}^\infty$ , where  $Q_n(\cdot)$  is a polynomial of degree  $n$ , which converges uniformly to  $f$  on the interval. If  $Q_n(x) = \sum_{i=0}^n \gamma_{n,i} x^i$ , let  $Q'_n(x_1, \dots, x_n) = \sum_{i=0}^n \gamma_{n,i} \prod_{j=1}^i x_j$ . Now, consider the estimator which draws a positive integer  $N$  according to some distribution  $\Pr(N = n) = p_n$ , samples  $X$  for  $N$  times to get  $x_1, x_2, \dots, x_N$ , and returns  $\frac{1}{p_N} (Q'_N(x_1, \dots, x_N) - Q'_{N-1}(x_1, \dots, x_{N-1}))$ , where we assume  $Q'_0 = 0$ . The expected value of this estimator is equal to:

$$\mathbb{E}_{N, x_1, \dots, x_N} \left[ \frac{(Q'_N(x_1, \dots, x_N) - Q'_{N-1}(x_1, \dots, x_{N-1}))}{p_N} \right] = \sum_{n=1}^{\infty} (Q_n(\mathbb{E}[X]) - Q_{n-1}(\mathbb{E}[X])) = f(\mathbb{E}[X]).$$

Thus, we have an unbiased estimator of  $f(\mathbb{E}[X])$ .

This technique appeared in a rather obscure early 1960's paper (Singh 1964) from sequential estimation theory, and appears to be little known outside of that community. However, we believe this technique is interesting, and expect it to have useful applications for other problems as well.

Getting back to our main problem of learning kernel-based predictors from noisy data, in (Cesa-Bianchi, Shalev-Shwartz, and Shamir 2010b) we propose a learning algorithm which is parameterized by a user-defined parameter  $p$ . This parameter allows one to perform a *tradeoff* between the number of noisy copies required for each example, and the total number of examples. In other words, the performance will be similar whether many noisy measurements are provided on a few examples, or just a few noisy measurements are provided on many different examples. Therefore, similarly to attribute efficient learning, we can compensate for partial information on each individual example (corresponding to a small number of noisy measurements) with the requirement of having a large number of different examples.

## Discussion

It is well-known that more examples can lead to more accurate predictors. In this work we study how more examples can also compensate for missing information on each individual example. The main idea of our technique is to model missing information as noise (sometimes, by using active exploration). By doing so, more data can reduce the noise variance, enabling us to compensate for the noise.

Other than the possible applications already discussed, additional reasons might prevent the learner from training on full information examples. For example, individual records in a medical database typically can not be fully disclosed for privacy reasons. In such situations, privacy-preserving techniques ensure that the information released about the data is enough to learn an accurate predictor, but not sufficient to recover any individual example —see (Chaudhuri and Monteleoni 2009). Again, the price to pay is the need to use a number of training examples larger than that required to learn a predictor with the same accuracy, but disregarding any privacy concerns.

There may be additional techniques that allows one to exploit the availability of many examples in a non-trivial way. For example, one can sometimes reduce the running time of a learning algorithm by replacing the original hypothesis class with a different, larger hypothesis class which has more structure. The price we need to pay is significantly more data, since from the statistical point of view, it is much harder to learn the new hypothesis class. This technique has been used in (Kakade, Shalev-Shwartz, and Tewari 2008) for the problem of online categorization with limited feedback.

## References

- Auer, P.; Cesa-Bianchi, N.; Freund, Y.; and Schapire, R. 2003. The nonstochastic multiarmed bandit problem. *SICOMP: SIAM Journal on Computing* 32.
- Cesa-Bianchi, N.; Shalev-Shwartz, S.; and Shamir, O. 2010a. Efficient learning with partially observed attributes. In *ICML*.
- Cesa-Bianchi, N.; Shalev-Shwartz, S.; and Shamir, O. 2010b. Online learning of noisy data with kernels. In *COLT*.
- Cesa-Bianchi, N.; Shalev-Shwartz, S.; and Shamir, O. 2010c. Some impossibility results for budgeted learning. In *Joint ICML-COLT workshop on Budgeted Learning*.
- Chaudhuri, K., and Monteleoni, C. 2009. Privacy-preserving logistic regression. In *NIPS*, 289–296.
- Cristianini, N., and Shawe-Taylor, J. 2004. *Kernel Methods for Pattern Analysis*. Cambridge University Press.
- Cun, Y. L. L.; Bottou, L.; Bengio, Y.; and Haffner, P. 1998. Gradient-based learning applied to document recognition. *Proceedings of IEEE* 86(11):2278–2324.
- Delalleau, O.; Bengio, Y.; and Roux, N. 2005. Efficient non-parametric function induction in semi-supervised learning. In *AISTATS*.
- Dempster, A.; Laird, N.; and Rubin, D. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Ser. B* 39:1–38.
- Devroye, L.; Györfi, L.; and Lugosi, G. 1996. *A Probabilistic Theory of Pattern Recognition*. Springer.
- Haussler, D. 1992. Decision theoretic generalizations of the PAC model for neural net and other learning applications. *Information and Computation* 100(1):78–150.
- Kakade, S.; Shalev-Shwartz, S.; and Tewari, A. 2008. Efficient bandit algorithms for online multiclass prediction. In *ICML*.
- Paninski, L. 2003. Estimation of entropy and mutual information. *Neural Computation* 15(6):1191–1253.
- Schölkopf, B., and Smola, A. J. 2002. *Learning with Kernels*. MIT Press.
- Singh, R. 1964. Existence of unbiased estimates. *Sankhyā: The Indian Journal of Statistics* 26(1):93–96.
- Vapnik, V. N. 1998. *Statistical Learning Theory*. Wiley.
- Zhu, X., and Lafferty, J. 2005. Harmonic mixtures: combining mixture models and graph-based methods for inductive and scalable semi-supervised learning. In *ICML*.
- Zhu, X. 2006. Semi-supervised learning literature survey.