, , 1–8 ()

A Note on Learning from Multiple-Instance Examples

AVRIM BLUM avrim+@cs.cmu.edu
ADAM KALAI akalai+@cs.cmu.edu
School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213

Abstract.
We describe a simple reduction from the problem of PAC-learning from multiple-instance examples to that of PAC-learning with one-sided random classification noise. Thus, all concept classes learnable with one-sided noise, which includes all concepts learnable in the usual 2-sided random noise model plus others such as the parity function, are learnable from multiple-instance examples. We also describe a more efficient (and somewhat technically more involved) reduction to the Statistical-Query model that results in a polynomial-time algorithm for learning axis-parallel rectangles with sample complexity $\tilde{O}(d^2r/\epsilon^2)$, saving roughly a factor of r over the results of Auer et al. (1997).

Keywords: Multiple-instance examples, classification noise, statistical queries

1. Introduction and Definitions

In the standard PAC learning model, a learning algorithm is repeatedly given labeled examples of an unknown target concept, drawn independently from some probability distribution. The goal of the algorithm is to approximate the target concept with respect to this distribution. In the multiple-instance example setting, introduced in (Dietterich et al., 1997), the learning algorithm is given only the following weaker access to the target concept: instead of seeing individually labeled points from the instance space, each "example" is an r-tuple of points together with a single label that is positive if *at least one* of the points in the r-tuple is positive (and is negative otherwise). The goal of the algorithm is to approximate the induced concept over these r-tuples. In the application considered by Dietterich et al., an example is a molecule and the points that make up the example correspond to different physical configurations of that molecule; the label indicates whether or not the molecule has a desired binding behavior, which occurs if at least one of the configurations has the behavior.

Formally, given a concept c over instance space X, let us define c_{multi} over X^* as:

$$c_{multi}(x_1, x_2, \ldots, x_r) \;=\; c(x_1) \vee c(x_2) \vee \ldots \vee c(x_r).$$

Similarly, given a concept class C, let $C_{multi} = \{c_{multi} : c \in C\}$. We will call $\vec{x} = (x_1, \ldots, x_r)$ an r-*example* or r-*instance*. Long and Tan (1996) give a natural PAC-style formalization of the multiple-instance example learning problem, which we may phrase as follows:

Definition 1. An algorithm A PAC-learns concept class C from multiple-instance examples if for any $r > 0$, and any distribution D over single instances, A PAC-learns C_{multi} over distribution D^r. (That is, each instance in each r-example is chosen independently from the same distribution D.)

Previous work on learning from multiple-instance examples has focused on the problem of learning d-dimensional axis-parallel rectangles. Dietterich et al. (1997) present several algorithms and describe experimental results of their performance on a molecule-binding domain. Long and Tan (1996) describe an algorithm that learns axis-parallel rectangles in the above PAC setting, under the condition that D is a product distribution (i.e., the coordinates of each single-instance are chosen independently), with sample complexity $\tilde{O}(d^2r^6/\epsilon^{10})$. Auer et al. (1997) give an algorithm that does not require D to be a product distribution and has a much improved sample complexity $\tilde{O}(d^2r^2/\epsilon^2)$ and running time $\tilde{O}(d^3r^2/\epsilon^2)$. (The $\tilde{O}$ notation hides logarithmic factors.) Auer (1997) reports on the empirical performance of this algorithm. Auer et al. also show that if we generalize Definition 1 so that the distribution over r-examples is arbitrary (rather than of the form D^r) then learning axis-parallel rectangles is as hard as learning DNF formulas in the PAC model.

In this paper we describe a simple general reduction from the problem of PAC-learning from multiple-instance examples to that of PAC-learning with one-sided random classification noise. Thus, all concept classes learnable from one-sided noise are PAC-learnable from multiple-instance examples. This includes all classes learnable in the usual 2-sided random noise model, such as axis-parallel rectangles, plus others such as parity functions. We also describe a more efficient reduction to the Statistical-Query model (Kearns, 1993). For the case of axis-parallel rectangles, this results in an algorithm with sample complexity $\tilde{O}(d^2r/\epsilon^2)$, saving roughly a factor of r over the results in (Auer et al., 1997).

2. A simple reduction to learning with noise

Let us define 1-sided random classification noise to be a setting in which positive examples are correctly labeled but negative examples have their labels flipped with probability $\eta < 1$, and the learning algorithm is allowed time polynomial in $\frac{1}{1-\eta}$.

THEOREM 1 *If C is PAC-learnable from 1-sided random classification noise, then C is PAC-learnable from multiple-instance examples.*

COROLLARY 1 *If C is PAC-learnable from (2-sided) random classification noise, then C is learnable from multiple-instance examples. In particular, this includes all classes learnable in the Statistical Query model.*

Proof (of Theorem 1 and Corollary 1): Let D be the distribution over single instances, so each multiple-instance example consists of r independent draws from D. Let p_{neg} be the probability a single instance drawn from D is a negative example of target concept c. So, a multiple-instance example has probability $q_{neg} = (p_{neg})^r$ of being labeled negative. Let $\hat{q}_{neg}$ denote the fraction of observed multiple-instance examples labeled negative; i.e., $\hat{q}_{neg}$ is the observed estimate of q_{neg}. Our algorithm will begin by drawing $O(\frac{1}{\epsilon}\log\frac{1}{\delta})$ examples and halting with the hypothesis "all positive" if $\hat{q}_{neg} < 3\epsilon/4$. Chernoff bounds guarantee that if $q_{neg} < \epsilon/2$ then with high probability we will halt at this stage, whereas if $q_{neg} > \epsilon$ then with high probability we will not. So, from now on we may assume without loss of generality that $q_{neg} \geq \epsilon/2$.

Given a source of multiple-instance examples, we now convert it into a distribution over single-instance examples by simply taking the first instance from each example and ignoring the rest. Notice that the instances produced are distributed independently according to D and for each such instance x,

- if x is a true positive, it is labeled positive with probability 1,
- if x is a true negative, it is labeled negative with probability $(p_{neg})^{r-1}$, independent of the other instances and labelings in the filtered distribution.

Thus, we have reduced the multiple-instance learning problem to the problem of learning with 1-sided classification noise, with noise rate $\eta = 1 - (p_{neg})^{r-1}$. Furthermore, η is not too close to 1, since

$$\eta \;=\; 1 - (p_{neg})^{r-1} \;\leq\; 1 - q_{neg} \;\leq\; 1 - \epsilon/2.$$

We can now reduce this further to the more standard problem of learning from 2-sided noise by independently flipping the label on each positive example with probability $\nu = \eta/(1+\eta)$ (that is, the noise rate on positive examples, ν, equals the noise rate on negative examples, $\eta(1-\nu)$). This results in 2-sided random classification noise with noise rate

$$\nu \;\leq\; (1-\epsilon/2)/(2-\epsilon/2) \;\leq\; 1/2 - \epsilon/8.$$

This reduction to 2-sided noise nominally requires knowing η; however, there are two easy ways around this. First, if there are m_+ positive examples, then for each $i \in \{0, 1, \ldots, m_+\}$ we can just flip the labels on a random subset of i positive examples and apply our 2-sided noise algorithm, verifying the m_+ hypotheses produced on an independent test set. The desired experiment of flipping each positive label with probability ν can be viewed as a probability distribution over these m_+ experiments, and therefore if the class is learnable with 2-sided noise then at least one of these will succeed. A second approach is that we in fact do have a good guess for η: $\eta = 1 - (q_{neg})^{1-1/r}$, so $\hat{\eta} = 1 - (\hat{q}_{neg})^{1-1/r}$ provides a good estimate for sufficiently large sample sizes. We discuss the details of this approach in the next section.

Finally, notice that it suffices to approximate c to error ϵ/r over single instances to achieve an ϵ-approximation over r-instances. ■

While we can reduce 1-sided noise to 2-sided noise as above, 1-sided noise appears to be a strictly easier setting. For instance, the class of *parity* functions, not known to be learnable with 2-sided noise, is easily learnable with 1-sided noise because parity is learnable from negative examples only. In fact, we do not know of *any* concept class learnable in the PAC model that is not also learnable with 1-sided noise.

3. A more efficient reduction

We now describe a reduction to the Statistical Query model of Kearns (1993) that is more efficient than the above method in that all of the r single instances in each r-instance are used. Our reduction turns out to be simpler than the usual reduction from classification noise for two reasons. First of all, we have a good estimate of the noise just based on the observed fraction of negatively-classified r-instances, $\hat{q}_{neg}$. Secondly, we have a source of examples with known (negative) classifications.

Informally, a Statistical Query is a request for a statistic about labeled instances drawn independently from D. For example, we might want to know the probability that a random instance $x \in \Re$ is labeled negative and satisfies $x < 2$. Formally, a statistical query is a pair (χ, τ), where χ is a function $\chi : X \times \{0,1\} \rightarrow \{0,1\}$ and $\tau \in (0,1)$. The statistical query returns an approximation $\hat{P_\chi}$ to the probability $P_\chi = \Pr_{x \in D}[\chi(x, c(x)) = 1]$, with the guarantee that $P_\chi - \tau \leq \hat{P_\chi} \leq P_\chi + \tau$. We know, from Corollary 1, that anything learnable in the Statistical Query model can be learned from multiple instance examples. In this section we give a reduction which shows:

THEOREM 2 *Given any $\delta, \tau \in (0, 1/r)$ and a lower bound $\tilde{q}_{neg}$ on q_{neg}, we can use a multiple-instance examples oracle to simulate n Statistical Queries of tolerance τ with probability at least $1-\delta$, using $O(\frac{\ln(n/\delta)}{r\tau^2 \tilde{q}_{neg}})$ r-instances, and in time $O(\frac{\ln(n/\delta)}{\tau^2}(\frac{1}{\tilde{q}_{neg}} + nT_\chi))$, where T_χ is the time to evaluate a query.*

Proof: We begin by drawing a set R of r-instances. Let S_- be the set of single instances from the negative r-instances in R, and let $S_{+/-}$ be the set of single instances from *all* r-instances in R. Thus the instances in $S_{+/-}$ are drawn independently from D, and those in S_- are drawn independently from D^-, the distribution induced by D over the negative instances.

We now estimate $\hat{q}_{neg} = |S_-|/|S_{+/-}|$ and $\hat{p}_{neg} = (\hat{q}_{neg})^{1/r}$. Chernoff bounds guarantee that so long as $|R| \geq k \frac{\ln(1/\delta)}{r^2\tau^2\tilde{q}_{neg}}$ for sufficiently large constant k, with probability at least $1 - \delta/2$,

$$q_{neg}(1 - r\tau/12) \leq \hat{q}_{neg} \leq q_{neg}(1 + r\tau/6).$$

This implies

$$\begin{aligned} p_{neg}(1 - r\tau/12)^{1/r} &\leq \hat{p}_{neg} \leq p_{neg}(1 + r\tau/6)^{1/r}, \\ p_{neg}(1 - \tau/6) &\leq \hat{p}_{neg} \leq p_{neg}(1 + \tau/6) \end{aligned}$$

where the last line follows using the fact that $\tau/6 < 1/r$.

Armed with $S_{+/-}$, S_-, and $\hat{p}_{neg}$, we are ready to handle a query. Our method will be similar in style to the usual simulation of Statistical Queries in the 2-sided noise model (Kearns, 1993), but different in the details because we have 1-sided noise (and, in fact, simpler because we have an estimate $\hat{p}_{neg}$ of the noise rate). Observe that, for an arbitrary subset $S \subseteq X$, we can directly estimate $Pr_{x\in D}[x \in S]$ from $S_{+/-}$. Using examples from S_-, we can also estimate the quantity,

$$\begin{aligned} Pr_{x\in D}[x \in S \wedge c(x) = 0] &= Pr_{x\in D}[c(x) = 0] Pr_{x\in D}[x \in S | c(x) = 0] \\ &= p_{neg} Pr_{x\in D^-}[x \in S]. \end{aligned} \tag{1}$$

Suppose we have some query (χ, τ). Define two sets: X_0 consists of all points $x \in X$ such that $\chi(x, 0) = 1$, and X_1 consists of all points $x \in X$ such that $\chi(x, 1) = 1$. Based on these definitions and (1), we can rewrite P_χ,

$$\begin{aligned} P_\chi &= Pr_{x\in D}[x \in X_1 \wedge c(x) = 1] + Pr_{x\in D}[x \in X_0 \wedge c(x) = 0] \\ &= Pr_{x\in D}[x \in X_1] - Pr_{x\in D}[x \in X_1 \wedge c(x) = 0] \\ &\quad + Pr_{x\in D}[x \in X_0 \wedge c(x) = 0] \\ &= Pr_{x\in D}[x \in X_1] + p_{neg}(Pr_{x\in D^-}[x \in X_0] - Pr_{x\in D^-}[x \in X_1]). \end{aligned} \tag{2}$$

Each of the three probabilities in the last equation is easily estimated from $S_{+/-}$ or S_- as follows:

> Using $k\frac{\ln(n/\delta)}{\tau^2}$ examples from $S_{+/-}$, estimate $Pr_{x\in D}[x \in X_1]$.
> Using $k\frac{\ln(n/\delta)}{\tau^2}$ examples from S_-, estimate $Pr_{x\in D^-}[x \in X_0]$ and
> $Pr_{x\in D^-}[x \in X_1]$.
> Combine these with $\hat{p}_{neg}$ to get an estimate $\hat{P}_\chi$ for
> $P_\chi = Pr_{x\in D}[x \in X_1] + p_{neg}(Pr_{x\in D^-}[x \in X_0] - Pr_{x\in D^-}[x \in X_1])$.

We can choose k large enough so that, with probability at least $1 - \delta/2n$, our estimates for $Pr_{x\in D}[x \in X_1]$, $Pr_{x\in D^-}[x \in X_0]$, and $Pr_{x\in D^-}[x \in X_1]$ are all within an additive $\tau/6$ of their true values. From above, we already know that $\hat{p}_{neg}$ is within an additive $\tau/6$ of p_{neg}. Now, since we have an additive error of at most $\tau/6$ on all quantities in (2), and each quantity is at most 1, our error on P_χ will be at most $\tau/6 + (1 + \tau/6)(1 + 2\tau/6) - 1 < \tau$, with probability at least $1 - \delta$ for all n queries. The runtime for creating $S_{+/-}$ and S_- is $O(\frac{\ln(n/\delta)}{\tau^2 \hat{q}_{neg}})$ and for each query is $O(\frac{\ln(n/\delta)}{\tau^2} T_\chi)$. The total number of r-instances required is $O(\frac{\ln(n/\delta)}{r\tau^2 \hat{q}_{neg}})$. ∎

As noted in Section 2, if we can approximate the target concept over single instances to error ϵ/r, then we have an ϵ-approximation over multiple-instance examples. Again, if we begin by drawing $O(\frac{1}{\epsilon} \log \frac{1}{\delta})$ examples and halting with the hypothesis "all positive" if $\hat{q}_{neg} < 3\epsilon/4$, then we get (using the lower bound $\epsilon/2$ for q_{neg}),

COROLLARY 2 *Suppose C is PAC-learnable to within error ϵ/r with n statistical queries of tolerance $\tau < 1/r$, which can each be evaluated in time T_χ (so n, τ,* *and T_χ depend on ϵ/r). Then C is learnable from multiple-instance examples with probability at least $1-\delta$, using $O(\frac{\ln(n/\delta)}{r\epsilon\tau^2})$ r-instances, and in time $O(\frac{\ln(n/\delta)}{\tau^2}(\frac{1}{\epsilon}+nT_\chi))$.*

The following theorem (given in (Auer et al., 1997) for the specific case of axis-parallel rectangles) gives a somewhat better bound on the error we need on single-instance examples.

THEOREM 3 *If $q_{neg} \geq \frac{\epsilon}{4}$ and $error_D(c,h) < \frac{\epsilon}{r}\frac{p_{neg}}{4q_{neg}}$, then $error_{D^r}(c_{multi}, h_{multi}) < \epsilon$.*

Proof: Let $p_1 = \Pr_{x\in D}[c(x)=0 \vee h(x)=0]$ and $p_2 = \Pr_{x\in D}[c(x)=0 \wedge h(x)=0]$. So, $error_D(c,h) = p_1 - p_2$. Notice that $\Pr_{\vec{x}\in D^r}[c_{multi}(\vec{x}) = h_{multi}(\vec{x}) = 0] = p_2^r$. Also note that $\Pr_{\vec{x}\in D^r}[c_{multi}(\vec{x}) = h_{multi}(\vec{x}) = 1] \geq 1 - p_1^r$ because all r-instances that *fail* to satisfy this equality must have their components drawn from the region $[c(x)=0 \vee h(x)=0]$. Therefore,

$$\begin{aligned}
error_{D^r}(c_{multi}, h_{multi}) &\leq p_1^r - p_2^r \\
&= (p_1 - p_2)(p_1^{r-1} + p_1^{r-2}p_2 + \ldots + p_2^{r-1}) \\
&\leq (p_1 - p_2) r p_1^{r-1} \\
&< \left(\frac{\epsilon}{r}\frac{p_{neg}}{4q_{neg}}\right) r \left(p_{neg} + \frac{\epsilon}{r}\frac{p_{neg}}{4q_{neg}}\right)^{r-1} \\
&\leq \epsilon \frac{(p_{neg})^r}{4q_{neg}} \left(1 + \frac{\epsilon}{4rq_{neg}}\right)^{r-1} \\
&\leq \frac{\epsilon}{4}\left(1 + \frac{1}{r}\right)^{r-1} \\
&\leq \epsilon.
\end{aligned}$$

■

4. Axis-Parallel Rectangles

The d-dimensional axis-parallel rectangle defined by two points, $(a_1, \ldots, a_d)$ and $(b_1, \ldots, b_d)$, is $\{\vec{x} | x_i \in [a_i, b_i], i = 1, \ldots, d\}$. The basic approach to learning axis-parallel rectangles with statistical queries is outlined in (Kearns, 1993) and is similar to (Auer et al., 1997). Suppose we have some target rectangle defined by two points, $(a_1, \ldots, a_d)$ and $(b_1, \ldots, b_d)$, with $a_i < b_i$. Our strategy is to make estimates $(\hat{a}_1, \ldots, \hat{a}_d)$ and $(\hat{b}_1, \ldots, \hat{b}_d)$, with $\hat{a}_i \geq a_i$ and $\hat{b}_i \leq b_i$ so that our rectangle is contained inside the true rectangle but so that it is unlikely that any point has ith coordinate between a_i and $\hat{a}_i$ or between $\hat{b}_i$ and b_i. We assume in what follows that $\epsilon/2 \leq q_{neg} \leq 1 - \epsilon/2$, and that we have estimates of p_{neg} and q_{neg} good to within a factor of 2, which we may do by examining an initial sample of size $O(\frac{1}{\epsilon}\log\frac{1}{\delta})$.

Let $\tau = \frac{\epsilon}{8dr}\frac{p_{neg}}{q_{neg}}$. From Theorem 3, we see that if we have error less than τ per side of the rectangle, then we will have less than ϵ error for the r-instance problem, and we are done. For simplicity, the argument below will assume that τ is known; if desired one can instead use an estimate of τ obtained from sampling, in a straightforward way.

We first ask the statistical query $\Pr_{x\in D}[c(x)=1]$ to tolerance $\tau/3$. If the result is less than $2\tau/3$ then $1-p_{neg}\leq\tau$, and (using Theorem 3) we can safely hypothesize that all points are negative. Otherwise we know $p_{neg}\leq 1-\tau/3$. Define $(a'_1,\ldots,a'_d)$ and $(b'_1,\ldots,b'_d)$ such that $\Pr_{x\in D}(a_i\leq x_i\leq a'_i)=\tau/3$ and $\Pr_{x\in D}(b'_i\leq x\leq b_i)=\tau/3$. (If the distribution is not continuous, then let a'_i be such that $\Pr_{x\in D}(a_i\leq x_i<a'_i)\leq\tau/3$ and $\Pr_{x\in D}(a_i\leq x_i\leq a'_i)\geq\tau/3$, and similarly for b'_i.) We now explain how to calculate $\hat{a}_1$, for example, without introducing error of more than τ.

Take $m=O(\ln(d/\delta)/\tau)$ unlabeled sample points. With probability at least 1-$\delta/2d$, one of these points has its first coordinate between a_1 and a'_1 (inclusive) and let us assume this is the case. We will now do a binary search among the first coordinates of these points, viewing each as a candidate for $\hat{a}_1$ and asking the statistical query $\Pr_{x\in D}[c(x)=1\wedge x_1<\hat{a}_1]$ with tolerance $\tau/3$. If all of our $\log m$ queries are inside our tolerance, then we are guaranteed that *some* $\hat{a}_1\geq a_1$ will return a value at most $2\tau/3$. In particular, the *largest* such $\hat{a}_1$ is at least a_1 and satisfies $\Pr_{x\in D}[a_1\leq x_1<\hat{a}_1]\leq\tau$. We similarly find the other $\hat{a}_i$ and $\hat{b}_i$. We use the algorithm of Theorem 2 with confidence parameter $\delta'=\delta/(4d\log m)$ so that with probability at least $1-\delta/2$ none of our $2d\log m$ queries fail.

The total number of multiple-instance examples used is at most

$$O\left(\frac{m}{r}+\frac{\ln((2d\log m)/\delta')}{r\tau^2 q_{neg}}\right)=\tilde{O}\left(\frac{1}{r\tau^2 q_{neg}}\right)=\tilde{O}\left(\frac{d^2 r q_{neg}}{\epsilon^2 p_{neg}^2}\right)=\tilde{O}\left(\frac{d^2 r}{\epsilon^2}\right).$$

The time for the algorithm is the time to sort these m points plus the time for the $\log m$ calls per side of the rectangle, which by Theorem 2, is:

$$\begin{aligned}
&O\left(dm\log m+\frac{\ln((d\log m)/\delta')}{\tau^2 q_{neg}}+d\log m\frac{\ln((d\log m)/\delta')}{\tau^2}\right)\\
&=\ O\left(\frac{d^3r^2}{\epsilon^2}\log\left(\frac{dr}{\epsilon}\log\frac{1}{\delta}\right)\log\left(\frac{d}{\delta}\log\frac{r}{\epsilon}\right)\right)\\
&=\ \tilde{O}\left(\frac{d^3r^2}{\epsilon^2}\right).
\end{aligned}$$

This is almost exactly the same time bound as given in (Auer et al., 1997) except that they have an $\log(\frac{d}{\delta})$ instead of $\log(\frac{d}{\delta}\ln(\frac{r}{\epsilon}))$ for the last term. We use $\tilde{O}(rd^2/\epsilon^2)$ r-instances compared to $\tilde{O}(r^2d^2/\epsilon^2)$ r-instances.

Acknowledgments

We thank the Peter Auer and the anonymous referees for their helpful comments. This research was supported in part by NSF National Young Investigator grant CCR-9357793, a grant from the AT&T Foundation, and an NSF Graduate Fellowship.

References

Auer, P. (1997). On learning from multi-instance examples: Empirical evaluation of a theoretical approach. In *Proceedings of the Fourteenth International Conference on Machine Learning*.

Auer, P., Long, P., and Srinivasan, A. (1997). Approximating hyper-rectangles: Learning and pseudo-random sets. In *Proceedings of the 29th Annual ACM Symposium on Theory of Computing*. To appear.

Dietterich, T. G., Lanthrop, R. H., and Lozano-Perez, T. (1997). Solving the multiple-instance problem with axis-parallel rectangles. *Artifical Intelligence*, 89(1-2):31–71.

Kearns, M. (1993). Efficient noise-tolerant learning from statistical queries. In *Proceedings of the Twenty-Fifth Annual ACM Symposium on Theory of Computing*, pages 392–401.

Long, P. and Tan, L. (1996). PAC learning axis-aligned rectangles with respect to product distributions from multiple-instance examples. In *Proceedings of the 9th Annual Conference on Computational Learning Theory*, pages 228–234.