Learning Hurdles for Sleeping Experts

Varun Kanade VKANADE@FAS.HARVARD.EDU
Harvard University, Cambridge, MA 02138, USA

Thomas Steinke TSTEINKE@FAS.HARVARD.EDU
Harvard University, Cambridge, MA 20138, USA

Abstract

We study the online decision problem where the set of available actions varies over time, also called the *sleeping experts* problem. We consider the setting where the performance comparison is made with respect to the *best ordering* of actions in hindsight. In this paper, both the payoff function and the availability of actions is adversarial. Kleinberg et al. (2008) gave a computationally efficient no-regret algorithm in the setting where payoffs are stochastic. Kanade et al. (2009) gave an efficient no-regret algorithm in the setting where action availability is stochastic.

However, the question of whether there exists a *computationally efficient* no-regret algorithm in the adversarial setting was posed as an open problem by Kleinberg et al. (2008). We show that such an algorithm would imply an algorithm for PAC learning DNF, a long standing important open problem. We believe that such computational limitations, especially for non-stochastic contextual experts problems, are likely to exist and studying these will point to the (correct) semi-stochasticity assumptions that allow designing no-regret algorithms.

1. Introduction

In online decision problems, a decision-maker must choose one of n possible *actions*, in each of the total T rounds. The decision-maker receives a payoff in the range $[0, 1]$. In the *full information* or *expert* setting, at the end of each round, the decision-maker sees the payoff corresponding to each of the possible actions. In the *bandit* setting, she only observes the reward of the action that she chose. The goal of the decision maker is to maximize her total payoff across T rounds, or as is common in the *non-stochastic* setting, to minimize her *regret* with respect to a class of strategies. The regret of the decision-maker is defined as the difference between the payoff she would have received by following the best strategy in hindsight from the class and the payoff that she actually received.

In this paper, we focus on the so-called *sleeping experts* problem. In this problem, the set of actions available to the decision-maker at round t is a subset S^t of n possible actions. The class of strategies we compare against is the set of *rankings* over the n total actions. Each ranking induces a simple strategy for the online decision problem: pick the highest-ranked available action. As a motivating example, consider the problem of choosing an advertisement to display alongside a search query. Of all the ads that match the particular keyword, only a subset might be actually available for displaying because of budget, geographical or other constraints. In this case, we would like the decision-making algorithm to compare well against the best (in hindsight) hypothetical ranking on the ads.

Our work focuses on the fully non-stochastic setting, where both the set of available actions and their payoffs are decided by an adversary[1]. In this paper, we consider the case of an *oblivious adversary*, i.e. one that does not observe the actual (random) choices made by the decision-maker. Since, our results show computational difficulties in designing efficient no-regret algorithms, they are equally applicable to the more challenging case of an adaptive adversary. An algorithm that selects an action a^t at time step t is efficient, if it makes its choice (possibly using history) in time that is polynomial in n. An algorithm is said to

Appearing in *Proceedings of the 29th International Conference on Machine Learning*, Edinburgh, Scotland, UK, 2012.

[1] No-regret algorithms are known for the case when either the payoffs or action availabilities are stochastic; these are discussed in the related works section.

be a *no-regret* algorithm if its regret is $O(\text{poly}(n)T^{1-\delta})$ for some constant $\delta > 0$. An informal statement of our main result is:

Theorem 1. *If there exists a computationally efficient no-regret algorithm for the sleeping experts problem (with respect to ranking strategies), then the class of polynomial size* DNFs *is PAC-learnable under arbitrary distributions.*

In contrast to the above result, if computational efficiency is not a concern, it is easy to see that the Hedge algorithm (Freund & Schapire, 1995) achieves regret $O(\sqrt{n\log(n)T})$, by treating each of the $n!$ rankings as an expert. This observation was made by Kleinberg et al. (2008), who also show that when the class of online algorithms is restricted to those that select actions by sampling over rankings and *without* observing the set of available actions S^t, there is no efficient no-regret algorithm unless RP = NP. However, this is a severe restriction and whether there exists an efficient no-regret algorithm without such restrictions was posed by Kleinberg et al. as an open question. Our result shows that such an algorithm would imply an algorithm for PAC-learning DNFs under arbitrary distributions, a long standing important open problem. In fact, the best known algorithm for PAC-learning DNFs takes time $2^{\tilde{O}(n^{1/3})}$ (Klivans & Servedio, 2001).

Contextual Experts Setting: We observe that the sleeping experts problem may be regarded as a special setting of the contextual experts problem, where at each time step t, the context x^t made available to the decision-maker is the set of available (awake) experts. The contextual experts/bandit setting is particularly applicable to several practical on-line machine learning tasks. Recently, a result by Beygelzimer et al. (2011) showed that learning algorithms could be used to design no-regret contextual experts algorithm in the case when empirical risk minimization can be achieved. Our paper shows a result in the opposite direction that a contextual experts algorithm (if one exists) could be used to solve a supervised learning problem, in our case learning DNFs.

We note that under standard cryptographic assumptions it is easy to construct (possibly unnatural) contextual experts problems that do not have any computationally efficient no-regret algorithms. Our result shows that a natural problem, the sleeping experts problem, is also at least as hard as a well-known learning problem, PAC learning DNF expressions. We believe that under strong non-stochastic assumptions, i.e. an adversary decides the context and the payoffs, contextual experts problems may be as hard as *known* supervised learning problems. This insight may lead to the (correct) semi-stochastic assumptions, as in the case of sleeping experts, either the payoffs or action availability being determined stochastically.

Our contributions: The proof of our main result follows from the fact that online agnostic learning of disjunctions reduces to the sleeping experts problem. As far as we are aware, computational hardness assumptions have not been used to show lower bounds on regret in experts/bandits problems[2]. Lower bounds in the literature are usually based on information theoretic arguments (such as predicting coin tosses). In the sleeping experts setting, the information-theoretic lower bound of $\Omega(\sqrt{n\log(n)T})$ can indeed be achieved if computational efficiency is not a concern.

The set of available experts may be thought of as context information at each time step, and hence allows for encoding learning problems (in our case agnostic learning of disjunctions). We believe that such techniques may be applicable to other contextual experts/bandits problems. When not in the contextual experts/bandits setting, it is often possible to compete against a class of exponentially many experts (cf. (Cesa-Bianchi & Lugosi, 2006) Chap. 5).

Related Work: The most relevant related work to ours is that of Kleinberg et al. (2008) and Kanade et al. (2009). Kleinberg et al. showed that in the setting where payoffs are stochastic (i.e. are drawn from a fixed distribution on each round and independently for each action) and action availability is adversarial, there exists an efficient no-regret algorithm that is essentially information-theoretically optimal. Kanade et al. gave an efficient no-regret algorithm in the setting when the payoffs are set by an oblivious adversary, but the action availability is decided stochastically, i.e. a subset $S \subseteq [n]$ of available actions is drawn according to a fixed distribution at each time step. In contrast, our results in this paper show that an adversarial coupling between action availability and payoffs makes the problem much harder.

In earlier literature, different versions of the sleeping experts problems have been considered by Freund et al. (1997) and Blum & Mansour (2007). Our results are not applicable to their settings, and in fact computationally efficient no-regret algorithms are known in those settings.

Organization. In section 2, we formally define the sleeping experts problem and the gambling problem. Section 3 provides the relevant definitions of batch and

[2] In the case of online learning of concepts (such as linear separators), such lower bounds have been shown before (see e.g. (Shalev-Shwartz et al., 2010))

online agnostic learning. Section 4 contains the main reduction showing that the sleeping experts problem is at least as hard as PAC learning DNF.

2. Setting and Notation

Let $A = \{a_1, \ldots, a_n\}$ be the set of actions. Let T be the total number of time steps for the online decision problem. In the sleeping experts setting, at time step t, a subset $S^t \subseteq A$ of actions is available, from which the decision-maker picks an action $a^t \in S^t$. Let $p^t : S^t \to [0,1]$ be the payoff function, and for any action a, let $p^t(a)$ denote the payoff associated with action a at time step t. At the end of round t, the entire payoff function p^t is revealed to the decision-maker. The total payoff of the decision-maker across T rounds is simply:

$$P_{\text{DM}} = \sum_{t=1}^{T} p^t(a^t)$$

When choosing an action $a^t \in S^t$, at time step T, the decision-maker may use the history to guide her choice. If the adversary cannot see any of the choices of the decision-maker we say that the adversary is *oblivious*. An *adaptive* adversary can see the past choices of the decision-maker and may then decide the payoff function and action availability. In this paper, we only consider the oblivious adversarial setting, since the hardness of designing no-regret algorithms against oblivious adversaries also applies to the case of (stronger) adaptive adversaries. Also, we only consider the full information setting, since the bandit setting is strictly harder.

The set of *strategies* that the decision-maker has to compete against is defined by the set of *rankings* over the actions. Let Σ_A denote the set of all possible $n!$ rankings over the n total actions. Given a particular ranking $\sigma \in \Sigma_A$, the strategy is to play the highest ranked available action according to σ. For subset $S \subseteq A$ of available actions, let $\sigma(S) \in A$ denote the action in S which is ranked highest according to σ. Thus, the payoff obtained by playing according to strategy σ is:

$$P_\sigma = \sum_{t=1}^{T} p^t(\sigma(S^t))$$

The quantity of interest is the *regret* of the decision-maker with respect to the class of strategies defined by rankings. The regret is defined as the difference between the payoff that would have been attained by playing according to the *best ranking strategy in hindsight* and the actual payoff received by the decision maker. Thus,

$$\text{Regret}_{\text{DM}} = \max_{\sigma \in \Sigma_A} P_\sigma - P_{\text{DM}}$$

We say that an algorithm is no-regret, if by playing according to the algorithm the decision-maker can achieve regret $O(p(n)T^{1-\delta})$, where $p(n)$ is a polynomial in n and $\delta \in (0, 1/2]$. Furthermore, we say that such an algorithm is *computationally efficient*, if at each time step t, given the set S^t of available actions (and possibly using history), it selects an action $a^t \in S^t$ in time polynomial in n.

3. Agnostic Learning

In this section, we define online and batch agnostic learning. Let X be an instance space and n be a parameter than captures the representation size of X (e.g. $X = \{0,1\}^n$ or $X = \mathbb{R}^n$).

Online Agnostic Learning. The definition of online agnostic learning used here is slightly different to those previously used in the literature (cf. (Ben-David et al., 2009)), but is essentially equivalent. Our definition simplifies the presentation of our results.

An online agnostic learning algorithm observes examples one at a time; at time step t it sees example $\mathbf{x}^t$, makes a prediction $\hat{y}^t \in \{0,1\}$ (possibly using history) and then observes y^t. Let $s = \langle(\mathbf{x}^t, y^t)\rangle_{t=1}^T$ be a sequence of length T, where $\mathbf{x}^t \in X$ and $y^t \in \{0,1\}$. We consider the oblivious adversarial setting, where the sequence s may be fixed by an adversary but is fixed ahead of time, i.e. without observing the past predictions made by the online learning algorithm. We define error of an online agnostic learning algorithm $\mathcal{A}$ with respect to a sequence $s = \langle(\mathbf{x}^t, y^t)\rangle_{t=1}^T$ as:

$$\text{err}_s(\mathcal{A}) = \frac{1}{T}\sum_{t=1}^{T} \mathbb{I}(\hat{y}^t \neq y^t)$$

where $\mathbb{I}$ is the indicator function. For any boolean function $f : X \to \{0,1\}$ we can define error of f with respect to the sequence $s = \langle(\mathbf{x}^t, y^t)\rangle_{t=1}^T$ as,

$$\text{err}_s(f) = \frac{1}{T}\sum_{t=1}^{T} \mathbb{I}(f(\mathbf{x}^t) \neq y^t).$$

For a concept class C of boolean functions over X, online agnostic learnability of C is defined as[3]:

Definition 2 (Online Agnostic Learning)**.** *We say that a concept class C over X is online agnostically learnable if there exists an online agnostic learning algorithm $\mathcal{A}$, that for all T, for all example sequences*

[3]The definition assumes that the online algorithm is deterministic; one may instead also allow a randomized algorithm that achieves low regret with high probability over its random choices. But, the guarantee must hold with respect to all sequences.

$s = \langle(\mathbf{x}^t, y^t)\rangle_{t=1}^T$, makes predictions $\hat{y}^1, \ldots, \hat{y}^T$ such that,

$$\mathrm{err}_s(\mathcal{A}) \leq \min_{f \in C} \mathrm{err}_s(f) + O(p(n)/T^\zeta)$$

for some polynomial $p(n)$ and $\zeta \in (0, 1/2]$. Furthermore, the running time of $\mathcal{A}$ at each time step must be polynomial in n. We say that $\mathcal{A}$ has regret bound *$O(p(n)/T^\zeta)$.*

Batch Agnostic Learning. We also give a definition of (batch) agnostic learning (cf. (Haussler, 1992), (Kearns et al., 1994)). For a distribution D over $X \times \{0,1\}$ and any boolean function $f : X \to \{0,1\}$ define,

$$\mathrm{err}_D(f) = \Pr_{(x,y)\sim D}[f(x) \neq y]$$

Definition 3 ((Batch) Agnostic Learning (Kearns et al., 1994))**.** *We say that a concept class C is (batch) agnostically learnable, if there exists an efficient algorithm that for every $\epsilon, \delta > 0$ and for every distribution D over $X \times \{0,1\}$, with access to a random example oracle from D, with probability at least $1-\delta$ outputs a hypothesis h such that,*

$$\mathrm{err}_D(h) \leq \min_{f \in C} \mathrm{err}_D(f) + \epsilon$$

The running time of the algorithm is polynomial in $n, 1/\epsilon, 1/\delta$ and h is polynomially evaluatable. The sample complexity *of the algorithm is the number of times it queries the example oracle.*

In most learning settings, it is well-known that batch learning is no harder than online learning. Theorem 4 follows more or less directly from (Littlestone, 1989; Cesa-Bianchi et al., 2004), but we provide a proof in Appendix A for completeness. Roughly speaking after an online to batch conversion, the sample complexity of the resulting batch algorithm is the number of time steps required to make the regret $O(\epsilon)$.

Theorem 4. *If a concept class C is online agnostically learnable with regret bound $O(p(n)/T^\zeta)$ then it is (batch) agnostically learnable. Furthermore the sample complexity for (batch) agnostic learning is $O((p(n)/\epsilon)^{1/\zeta}) + O(1/\varepsilon^4 + \log^2(1/\delta) + (1/\zeta\varepsilon^2)\log(n/\varepsilon\delta))$.*

4. Sleeping Experts Problem

In this section, we show that the sleeping experts problem is at least as hard as online agnostic learning of disjunctions. Theorem 4 implies that the class of disjunctions is also (batch) agnostically learnable. It is known that agnostic learning of disjunctions implies PAC learning of DNF expressions (cf. (Kearns et al., 1994; Kalai et al., 2009))[4], thus proving Theorem 1.

Recall that in the sleeping experts setting we consider, the action availability and payoff functions are set by an oblivious adversary. First, we define the notation used in this section. Let $X = \{0,1\}^n$ and let DISJ denote the class of disjunctions over X. Let $\mathbf{x} = x_1 \cdots x_n \in X$; for each bit x_i we define two actions O_i (corresponding to $x_i = 1$) and Z_i (corresponding to $x_i = 0$). We define an additional action $\perp$. Thus, the set of actions is $A = \{\perp, O_1, Z_1, \ldots, O_i, Z_i, \ldots, O_n, Z_n\}$.

Suppose there exists an algorithm Alg for the sleeping experts problem, that achieves regret $O(p(n)T^{1-\delta})$ for some polynomial $p(n)$ and $\delta \in (0, 1/2]$. We use Alg to construct an online learning algorithm DISJ-Learn (see Fig. 1) for online agnostic learning DISJ that has average regret $O(p(n)/T^\delta)$. The instance $\mathbf{x}^t$ is used to define the set of available actions at round t and the label y^t to define the payoffs.

Proposition 5. *Suppose there exists an efficient algorithm for the sleeping experts problem with regret $O(p(n)T^{1-\delta})$, then there exists an efficient online agnostic algorithm for learning disjunctions with average regret $O(p(n)/T^\delta)$.*

Proof. Let $s = \langle(\mathbf{x}^t, y^t)\rangle_{t=1}^T$ be any sequence of examples from $X \times \{0,1\}$ for the problem of online agnostic learning DISJ. Suppose there is an efficient algorithm Alg for the sleeping experts problem with regret $O(p(n)T^{1-\delta})$. Then, we claim that Algorithm DISJ-Learn (Fig. 1) has regret $O(p(n)/T^\delta)$.

Let the total set of actions be $A = \{\perp, O_1, Z_1, \ldots, O_n, Z_n\}$, Σ_A the set of rankings over A. Let the payoff functions, p^t, and the set of available actions, S^t, be as defined in Fig. 1. Let σ^* be the best ranking in hindsight, i.e. $\sigma^* = \mathrm{argmax}_{\sigma \in \Sigma_A} \sum_{t=1}^T P_\sigma$. Also, let f^* be the best disjunction with respect to the sequence s, i.e. $f^* = \mathrm{argmin}_{f \in \mathsf{DISJ}} \mathrm{err}_s(f)$.

Note that $\hat{y}^t$ is the prediction made by DISJ-Learn using the action selected by Alg. At any given round, the payoff received by Alg is $1 - \mathbb{I}(\hat{y}^t \neq y^t)$ (if Alg picks $\perp$, then payoff is $1 - y^t$ and $\hat{y}^t = 0$; otherwise, payoff is y^t and $\hat{y}^t = 1$). Hence, summing over all rounds,

[4] Actually, agnostically learning conjunctions implies PAC learning DNF, but because of the duality between conjunctions and disjunctions, an agnostic learning algorithm for learning disjunctions also implies an algorithm for earning conjunctions.

Algorithm. DISJ-Learn (Online Agnostic Learning)

Input: Alg - the algorithm for sleeping experts prob.

For $t = 1, \ldots, T$,

1. Receive example $\mathbf{x}^t$. Define $S^t = \{\perp\} \cup \{O_i \mid x_i^t = 1\} \cup \{Z_i \mid x_i^t = 0\}$.
2. Give S^t as the set of available actions to Alg. Let Alg choose a^t.
3. If $a^t = \perp$, then set $\hat{y}^t = 0$, else set $\hat{y}^t = 1$.
4. Observe y^t. Define $p^t(\perp) = 1 - y^t$ and $p^t(a) = y^t$ for all other actions $a \in S^t \setminus \{\perp\}$. Return p^t as the payoff function to Alg.

Figure 1. Algorithm for online agnostically learning DISJ.

$$\frac{1}{T} P_{\mathsf{Alg}} = 1 - \mathrm{err}_s(\mathsf{DISJ\text{-}Learn})$$

Now, the proof follows immediately from Lemma 6, since $1 = \min_{f \in \mathsf{DISJ}} \mathrm{err}_s(f) + (1/T) \max_{\sigma \in \Sigma_A} P_\sigma$, and hence from the above equation we get,

$$\begin{aligned} \mathrm{err}_s(\mathsf{DISJ\text{-}Learn}) - \min_{f \in \mathsf{DISJ}} \mathrm{err}_s(f) &= \frac{1}{T} \left(\max_{\sigma \in \Sigma_A} P_\sigma - P_{\mathsf{Alg}} \right) \\ &= O(p(n)/T^\delta). \end{aligned}$$

□

Lemma 6. *Let $s = \langle (\mathbf{x}^t, y^t) \rangle_{t=1}^T$ be any sequence of examples from $X \times \{0,1\}$. Let $A = \{\perp, O_1, Z_1, \ldots, O_n, Z_n\}$, Σ_A be the set of rankings over A and let S^t and p^t be as defined in Fig. 1. Then*

$$\min_{f \in \mathsf{DISJ}} \mathrm{err}_s(f) + \frac{1}{T} \max_{\sigma \in \Sigma_A} P_\sigma = 1$$

where P_σ is the payoff achieved by playing the sleeping experts problem according to ranking strategy σ.

Proof. Let σ be a ranking over the set of actions $A = \{\perp, O_1, Z_1, \ldots, O_n, Z_n\}$. For any two actions $a_1, a_2 \in A$, define $a_1 \prec_\sigma a_2$ to mean that a_1 is ranked higher by σ than a_2. For a ranking σ define a disjunction f_σ as:

$$f_\sigma = \bigvee_{i : O_i \prec_\sigma \perp} x_i \vee \bigvee_{i : Z_i \prec_\sigma \perp} \bar{x}_i$$

If for some i, both $O_i \prec_\sigma \perp$ and $Z_i \prec_\sigma \perp$, then $f_\sigma \equiv 1$. Note that several permutations may map to the same disjunction, since only which O_i and Z_i are ranked above $\perp$ is important, not their ranking relative to each other. We show that,

$$\mathrm{err}_s(f_\sigma) + \frac{1}{T} P_\sigma = 1 \tag{1}$$

Consider some vector $\mathbf{x}^t = \{0,1\}^n$ and let $S^t \subseteq A$ be the corresponding subset of available actions (see Fig. 1). Then, note that $f_\sigma(\mathbf{x}^t) = 0$ if and only if $\sigma(S^t) = \perp$. If the true label is $y^t = 1$, f_σ suffers error $1 - f_\sigma(\mathbf{x}^t)$ and $\sigma(S^t)$ receives payoff $f_\sigma(\mathbf{x}^t)$. If the true label is $y^t = 0$, then f_σ suffers error $f_\sigma(\mathbf{x}^t)$ and $\sigma(S^t)$ receives payoff $1 - f_\sigma(\mathbf{x}^t)$. Summing over $(\mathbf{x}^t, y^t)$ in the sequence s, we get (1). But, this also completes the proof of the lemma, since for every disjunction g there exists a ranking π such that $g = f_\pi$, e.g. the ranking where the actions corresponding to literals occurring in g (O_i or Z_i depending on whether x_i or $\bar{x}_i$ appears in g) are place first, followed by $\perp$, followed by the rest of the actions. □

Notes

A longer version of these appeared in the Proceedings of the conference, Innovations in Theoretical Computer Science (ITCS), 2012. The copyright to that version are owned by ACM.

Acknowledgments

Varun Kanade was supported in part by NSF grants CCF-04-27129 and CCF-09-64401. Thomas Steinke was supported in part by the Lord Rutherford Memorial Research Fellowship and NSF grant CCF-1116616. The authors would like to thank Adam Kalai, Tal Moran, Justin Thaler, Jonathan Ullman, Salil Vadhan, and Leslie Valiant for helpful discussions.

References

Ben-David, S., Pál, D., and Shalev-Shwartz, S. Agnostic online learning. In *COLT*, 2009.

Beygelzimer, Alina, Langford, John, Li, Lihong, Reyzin, Lev, and Schapire, Robert E. Contextual bandit algorithms with supervised learning guarantees. In *AISTATS*, 2011.

Blum, A. and Mansour, Y. From external to internal regret. *Journal of Machine Learning Research*, 8: 1307–1324, 2007.

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

Cesa-Bianchi, N., Conconi, A., and Gentile, C. On the generalization ability of on-line learning algorithms.

IEEE Transactions on Information Theory, 50(9): 2050–2057, 2004.

Freund, Y. and Schapire, R. E. A decision-theoretic generalization of on-line learning and an application to boosting. In *Proceedings of the second European conference on computational learning theory*, 1995.

Freund, Y., Schapire, R. E., Singer, Y., and Warmuth, M. K. Using and combining predictors that specialize. In *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*, 1997.

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

Kalai, A. T., Kanade, V., and Mansour, Y. Reliable agnostic learning. In *COLT*, 2009.

Kanade, V., McMahan, B., and Bryan, B. Sleeping experts and bandits with stochastic action availability and adversarial rewards. In *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics*, pp. 272–279, 2009.

Kearns, M. J., Schapire, R. E., and Sellie, L. M. Toward efficient agnostic learning. *Machine Learning*, 17(2-3):115–141, 1994.

Kleinberg, R., Niculescu-Mizil, A., and Sharma, Y. Regret bounds for sleeping experts and bandits. *Machine learning*, pp. 1–28, 2008.

Klivans, Adam R. and Servedio, Rocco. Learning dnf in time. In *Proceedings of the thirty-third annual ACM symposium on Theory of computing*, STOC '01, pp. 258–265, New York, NY, USA, 2001. ACM. ISBN 1-58113-349-9. doi: http://doi.acm.org/10.1145/380752.380809. URL http://doi.acm.org/10.1145/380752.380809.

Littlestone, N. From on-line to batch learning. In *Proceedings of the second annual workshop on computational learning theory*, 1989.

Shalev-Shwartz, S., Shamir, O., and Sridharan, K. Learning kernel-based halfspaces with the zero-one loss. In *Proceedings of the 23rd Annual Conference on Learning Theory*, 2010.

A. On-line to Batch Learning

We prove Theorem 4 using the following lemma.

Lemma 7. *Let $\mathcal{A}$ be an online agnostic learning algorithm for a concept class $\mathcal{C}$ over X with regret bound $O(p(n)/T^{\zeta})$. We run $\mathcal{A}$ for T steps on $s = \langle(\mathbf{x}^t, y^t)\rangle_{t=1}^T$. At each step $\mathcal{A}$ can be interpreted as a hypothesis H_s^t which computes $\hat{y}^t = H_s^t(\mathbf{x}^t)$.*

Then we can choose $T = O\left(p(n)/\varepsilon\right)^{1/\zeta} + O\left(1/\varepsilon^4 + \log^2(1/\delta)\right)$ such that the following holds.

Let D be a distribution over $X \times \{0,1\}$. Take a sequence $s = \langle(\mathbf{x}^t, y^t)\rangle_{t=1}^T$ of T examples from D. Let $\langle H^t\rangle_{t=1}^T$ be the hypotheses produced by $\mathcal{A}$ running on s. Then, with probability $1-\delta$ over the choice of s, there exists t^* such that $\operatorname{err}_D(H_s^{t^*}) \le \min_{f\in C}\operatorname{err}_D(f) + \varepsilon$.

Lemma 7 allows us to convert an online agnostic learning algorithm into a hypothesis, which we can use for (batch) agnostic learning.

Proof. Let $Q_s^t = \sum_{t'\le t}\operatorname{err}(H_s^{t'})$. Then, clearly, $\langle Q_s^t\rangle_{t=1}^T$ is a submartingale. Moreover, induction on T gives

$$\begin{aligned}\mathbb{E}_s[\operatorname{err}_s(\mathcal{A})] &= \mathbb{E}_s\left[\frac{1}{T}\sum_{t=1}^T \mathbb{I}(H_s^t(\mathbf{x}^t) \neq y^t)\right] \\ &= \mathbb{E}_s\left[\frac{1}{T}\sum_{t=1}^T \operatorname{err}_D(H_s^t)\right] = \mathbb{E}_s\left[\frac{Q_s^T}{T}\right]. \quad (2)\end{aligned}$$

We will now use standard bounds to show that (i) the expectation (2) is close to (or better than) the optimal error and that (ii) Q_s^T is close to its expectation with high probability. It follows that at least one hypothesis $H_s^{t^*}$ must have error close to (or better than) an optimal concept.

(i) We have

$$\begin{aligned}\mathbb{E}_s[\operatorname{err}_s(\mathcal{A})] &\le \mathbb{E}_s[\min_{f\in\mathcal{C}}\operatorname{err}_s(f)] + O(p(n)/T^{\zeta}) \\ &\le \min_{f\in\mathcal{C}}\operatorname{err}_D(f) + O(p(n)/T^{\zeta}). \quad (3)\end{aligned}$$

The first inequality follows from $\mathcal{A}$ being an online agnostic learning algorithm. The second inequality follows from the fact that $\mathbb{E}_s[\min_{f\in\mathcal{C}}\operatorname{err}_s(f)] \le \min_{f\in\mathcal{C}}\mathbb{E}_s[\operatorname{err}_s(f)]$.

(ii) Noting that $Q_s^t \le Q_s^{t+1} \le Q_s^t + 1$, Azuma's inequality gives

$$\Pr_s\left[Q_s^T \ge \mathbb{E}[Q_s^T] + T^{1-\alpha}\right] \le \exp\left(-T^{1-2\alpha}/2\right). \quad (4)$$

Combining (2), (3), and (4), we have

$$\begin{aligned}\Pr_s\left[\frac{1}{T}\sum_{t=1}^T \operatorname{err}_D(H_s^t) \ge \min_{f\in\mathcal{C}}\operatorname{err}_D(f) + T^{-\alpha} + O(p(n)/T^{\zeta})\right] \\ \le \exp\left(-T^{1-2\alpha}/2\right).\end{aligned}$$

So we can choose $\alpha = 1/4$ and

$$T = \max\left\{(2/\varepsilon)^4, O(2p(n)/\varepsilon)^{1/\zeta}, (2\log(1/\delta))^2\right\}$$

to ensure that, with probability $1-\delta$,

$$\min_{t=1}^{T} \mathrm{err}_D(H_s^t) \leq \frac{1}{T}\sum_{t=1}^{T} \mathrm{err}_D(H_s^t) \leq \min_{f\in\mathcal{C}} \mathrm{err}_D(f) + \varepsilon.$$

□

Proof of Theorem 4. Let $\mathcal{A}$ be an online agnostic learning algorithm for a concept class $\mathcal{C}$ over X with regret bound $O(p(n)/T^{\zeta})$. Fix $\varepsilon, \delta > 0$ and a distribution D over $X \times \{0,1\}$. Choose $T = O\left(p(n)/\varepsilon\right)^{1/\zeta} + O\left(1/\varepsilon^4 + \log^2(1/\delta)\right)$ as in Lemma 7. We sample $s = \langle(\mathbf{x}^t, y^t)\rangle_{t=1}^{T}$ from D and run $\mathcal{A}$ on s. Now we have a sequence of hypotheses $\langle H^t\rangle_{t=1}^{T}$. With probability $1-\delta/2$ over the choice of s, at least one hypothesis $H_s^{t^*}$ satisfies $\mathrm{err}_D(H^{t^*}) \leq \min_{f\in\mathcal{C}} \mathrm{err}_D(f) + \varepsilon/2$. All that remains is to identify one such hypothesis.

Take T' samples $s' = \langle(\mathbf{x}^{t'}, y^{t'})\rangle_{t'=1}^{T'}$ from D. By the Chernoff bound, for any $f : X \to \{0,1\}$,

$$\Pr_{s'}\left[|\mathrm{err}_{s'}(f) - \mathrm{err}_D(f)| \geq \varepsilon/2\right] \leq 2e^{-T'\varepsilon^2/16}.$$

Let $T' = (16/\varepsilon^2)\log(4T/\delta)$. Then

$$\Pr_{s'}\left[\forall t \;\; |\mathrm{err}_{s'}(H_s^t) - \mathrm{err}_D(H_s^t)| < \varepsilon/2\right] \geq 1 - \delta/2.$$

So we can estimate the accuracy of each hypothesis and identify a good one. Thus we take $T + T'$ samples and, with probability $1-\delta$, we can find a good hypothesis. □