Experience-Efficient Learning in Associative Bandit Problems

Alexander L. Strehl STREHL@CS.RUTGERS.EDU
Chris Mesterharm MESTERHA@CS.RUTGERS.EDU
Michael L. Littman MLITTMAN@CS.RUTGERS.EDU
Haym Hirsh HIRSH@CS.RUTGERS.EDU
Department of Computer Science, Rutgers University, Piscataway, NJ 08854 USA

Abstract

We formalize the associative bandit problem framework introduced by Kaelbling as a learning-theory problem. The learning environment is modeled as a k-armed bandit where arm payoffs are conditioned on an observable input selected on each trial. We show that, if the payoff functions are constrained to a known hypothesis class, learning can be performed efficiently with respect to the VC dimension of this class. We formally reduce the problem of PAC classification to the associative bandit problem, producing an efficient algorithm for any hypothesis class for which efficient classification algorithms are known. We demonstrate the approach empirically on a scalable concept class.

1. Introduction

Reinforcement learning is the problem of using experience to guide action selection to maximize utility. To do so, three challenging problems must be solved:

- **Temporal credit assignment**: Actions have an immediate cost or benefit, but may also affect utility by changing the state of the world and therefore opportunities for future benefits.
- **Exploration/exploitation**: Action selection influences utility gathering, but also opportunities for experience gathering—a reinforcement learner chooses when to learn.
- **Generalization**: A learner should use its experience in similar states to minimize the amount of redundant exploration it carries out.

Appearing in *Proceedings of the 23rd International Conference on Machine Learning*, Pittsburgh, PA, 2006.

Ultimately, we seek algorithms with formal guarantees that address all three challenges at once. As a step in this direction, we ignore the temporal credit-assignment problem in this paper and study a problem that requires exploration/exploitation and generalization. Although we devised a provably correct algorithm for the problem, we found that, under the PAC-style performance metric we analyzed, a pure exploration-based approach was sufficient to identify a near-optimal policy. More sophisticated exploration methods are required for a regret-based metric and probably for extensions of the problem setting.

Section 2 defines the associative bandit problem. Section 3 defines the associative-prediction problem and shows how its solution solves associative bandit problems; the section also provides efficient solutions to two special cases. Section 4 relates both problems to classification learning, Section 6 demonstrates the main ideas of the paper using a simple scalable learning problem, and Section 7 summarizes the results and concludes.

2. The Associative Bandit Problem

The *associative bandit problem* (ABP)[1] (Kaelbling, 1994) is a generalization of the k-armed bandit problem (Berry & Fristedt, 1985). The setting consists of an input space X, an unknown distribution D over X, a hypothesis space H (made up of functions from X into the set $\{1, 0\}$), a set of k arms, and for each arm i, an unknown *target* hypothesis $h_i^* \in H$, and two unknown payoff probabilities p_i^1 and p_i^0.

A learning sequence proceeds as follows. During each timestep, an input $x_t \in X$ is drawn according to the distribution D and is provided to the algorithm, which

[1] Kaelbling (1994) calls the problem "Associative Reinforcement Learning", but this term has been used inconsistently in the literature, so we chose a new term with a clearer derivation.

then selects an arm a_t and receives a Boolean payoff r_t. With probability $p_{a_t}^{h^*_{a_t}(x_t)}$, the payoff r_t will be 1, and with probability $1-p_{a_t}^{h^*_{a_t}(x_t)}$, it will be 0. The goal of the algorithm is to produce, after a finite number of steps, a policy $\hat{\pi} : X \to \{1, \ldots, k\}$ (a mapping from inputs to arms) that maximizes expected payoff. If an algorithm $\mathcal{A}$ terminates after m steps, we say that m is the *sample complexity* of $\mathcal{A}$ for that sequence.

Let π^* denote an optimal policy. It will satisfy $\pi^*(x) = \text{argmax}_i\, p_i^{h^*_i(x)}$. Let $\texttt{Ret}(\pi)$ denote the expected average return of policy π, formally,

$$\texttt{Ret}(\pi) = E_{x\sim D}\Big(p_{\pi(x)}^{h^*_{\pi(x)}(x)}\Big). \tag{1}$$

Let $\texttt{dim}(H)$ be the Vapnik-Chervonenkis (VC) dimension of H (Mitchell, 1997). An algorithm with low sample complexity that consistently produces high-quality polices is considered an *efficient* learner. The following definition, motivated by the PAC definition of Valiant (1984), formalizes this notion:

Definition 1 *We say that a learning algorithm $\mathcal{A}$ is* **ABP-PAC** *(PAC for associative bandit problems) if, for any inputs $0 < \epsilon < 1$ and $0 < \delta < 1$, it has sample complexity of m and with probability at least $1-\delta$, returns a policy π such that $\texttt{Ret}(\pi) \geq \texttt{Ret}(\pi^*) - \epsilon$. Furthermore, m is bounded by a polynomial in the input parameters $1/\epsilon$, $1/\delta$, k, and $\texttt{dim}(H)$.*

Note that Definition 1 puts no restriction on the computational complexity of executing the learning algorithm or the returned policy. Nevertheless, we will only be interested in policies that can be evaluated quickly (in time $O(k)$), and prefer learning algorithms with polynomial complexity.

2.1. Related Work

The ABP problem is similar to the problem studied by Fiechter (1995) when H is finite, which is directly relevant to our results in Section 3.4. In his work, the payoff probability for an arm given input x is determined by a *decision list* over the hypothesis space H. The ABP problem can be viewed as a specialization of this problem where each decision list is limited to a single hypothesis of H. However, Fiechter's (1995) solution to the more general problem does not provide an efficient solution in our specialized case for two reasons. First, the sample complexity is quadratic in terms of the size of the hypothesis space and thus does not satisfy the conditions of Definition 1. Our method (see Section 3.4) has sample complexity that is logarithmic in the size of H. Even this approach is not efficient when the VC dimension is much smaller than $\ln(|H|)$. In this case, one must resort to the more sophisticated methods of Section 4.1 to reduce the problem to classification and then use empirical loss minimization as described in Section 5. Second, the policy produced by Fiechter's (1995) approach has a worst-case computational complexity of $O(k|H|)$, which is much higher than what we desire.

Auer (2000) and Abe et al. (2003) study a similar problem and obtain better regret bounds under the assumption that the payoff or payoff probability for an arm is a linear function of the input.

We have studied ABP in a somewhat off-line setting where the arms are pulled a polynomial number of times and a near-optimal policy must be learned. However, Fiechter (1997) has shown that a near optimal policy can be used to achieve near optimal payoff by adopting the near optimal policy for a polynomial number of trials (exploiting).

3. The Associative Prediction Problem

The *associative-prediction problem* (APP) is that of predicting the mean payoff rate from the input for a specially structured probabilistic payoff function. The (hidden) structure of the problem is that all inputs labeled identically by an unknown Boolean concept have identical payoff rates.

The problem setting consists of an input space X, an unknown distribution D over X, a hypothesis space H, an unknown target hypothesis $h^* \in H$, and two unknown payoff probabilities p^1 and p^0.

Let $EX := EX(h^*, p^1, p^0, D)$ be an oracle that when called produces a labeled pair (x, y) where x is distributed according to D and $y \in \{0, 1\}$ is 1 with probability $p^{h^*(x)}$ and 0 with probability $1 - p^{h^*(x)}$.

A learning sequence proceeds as follows. During each timestep, the algorithm calls EX to receive a labeled pair (x, y). After m calls, a hypothesis $h \in H$ and two real numbers, $0 \leq \hat{p}^1 \leq 1$ and $0 \leq \hat{p}^0 \leq 1$ are produced as output.

Let $APP(X, H, \epsilon, \delta, \mu, EX)$ denote any *efficient* learning algorithm for APP. It takes inputs $0 < \epsilon < 1$, $0 < \delta < 1$, and $0 < \mu < 1$. With probability at least $1 - \delta$, its outputs must satisfy:

$$\Pr_{x\sim D}\Big(|\hat{p}^{h(x)} - p^{h^*(x)}| > \epsilon\Big) \leq \mu. \tag{2}$$

In other words, the error in its final payoff predictions should be within ϵ of the true expected payoff on at least a $1-\mu$ weighted fraction of the inputs, with high

probability. Furthermore, the number of calls to EX, denoted $APPSC(X, H, \epsilon, \delta, \mu, EX)$, is bounded by a polynomial in $1/\epsilon$, $1/\delta$, $1/\mu$, and $\texttt{dim}(H)$.

3.1. Reduction to the Associative Bandit Problem

In this section, we show that a solution to APP can be used to solve ABP. Suppose we are given an instance of ABP and some efficient learner for APP. For each arm i, consider the oracle $EX_i := EX(h_i^*, p_i^1, p_i^0, D)$. In a single timestep, we can simulate a call to EX_i by drawing an instance x from D and selecting arm i to receive payoff y. For each arm i, we call the APP solver, $APP(X, H, \epsilon/4, \delta/k, \epsilon/(2k), EX_i)$, which produces a hypothesis h_i and probabilities $\hat{p}_i^1$ and $\hat{p}_i^0$. The outputs are used to create the policy $\hat{\pi}(x) = \text{argmax}_i \, \hat{p}_i^{h_i(x)}$, which we claim solves ABP.

With probability at least $1 - k(\delta/k) = 1 - \delta$, all calls to the APP learning algorithm return accurate predictions. When this event happens, on any given input x chosen according to D, the probability that all k predictions ($\hat{p}_i^{h_i(x)}$) are $\epsilon/4$-accurate is at least $1 - k(\epsilon/(2k)) = 1 - \epsilon/2$. On such an instance, choosing the arm that has the highest estimated payoff gives an expected payoff no more than $\epsilon/2$ from optimal. In the remaining trials, we can pessimistically assume the predictions are maximally erroneous. Therefore, $\texttt{Ret}(\pi^*) - \texttt{Ret}(\pi) \leq (1 - \epsilon/2)\epsilon/2 + \epsilon/2 \leq \epsilon$, as required. The total number of arm pulls is $\sum_{i=1}^{k} APPSC(X, H, \epsilon/4, \delta/k, \epsilon/(2k), EX_i)$, which is a polynomial in the relevant quantities.

Our method of reduction produces a solution to ABP by pulling each arm a fixed number of times. This algorithm is similar to the "naïve" algorithm for solving the k-armed bandit problem (Fong, 1995). Although it provides an adequate solution for this problem, we believe more sophisticated exploration strategies are needed in other problem settings.

3.2. Discussion

The associative-prediction problem is a special case of *learning a probabilistic concept* (Kearns & Schapire, 1990). This connection allows us to derive a lower bound on the sample complexity of APP, although we do not present the argument here due to lack of space. Viewing APP in this way also suggests a possible solution to the ABP. Specifically, we can reduce the problem to a special instance of *regression* and utilize *empirical loss minimization* on the space of all possible payoff functions. This method results in a solution with polynomial sample complexity (see Theorems 5.1 and 5.2 of Kearns and Schapire (1990)). Rather than take this approach, we chose to further reduce the problem to classification.

Our motivation for reducing the problem to classification is as follows. First, the payoff probabilities of each arm belong to one of two classes. Determining which class an input belongs to is, in essence, a classification problem; reducing to classification takes advantage of this structure. Second, the hypothesis set for the regression problem (the set of possible payoff functions) is larger than the set of hypotheses (H) for the classification problem. In fact, even when H is finite the set of possible payoff functions is infinite (since we allow the payoff probability for each class to be any real number between 0 and 1). This setting rules out solutions that test each hypothesis explicitly. Lastly, our reduction allows us to generate many algorithms for the APP model by using various classification algorithms (neural networks or support vector machines, say). To directly use most popular regression algorithms (for example, linear regression algorithms), a different set of model assumptions are needed. A fruitful extensions of this work may be to consider such a model to allow a direct application of regression.

3.3. Two Special Cases

This section addresses two special cases of APP. The first is when the probability of h^* classifying a given instance as 1 or 0 is tiny. The second is when the payoff probabilities p^1 and p^0 are very close to each other. We claim that in these cases any hypothesis can be turned into a solution to APP. Thus, from now on we make the two following assumptions (defining $L = \mu\epsilon/12$):

$$\Pr_{x \sim D}(h^*(x) = 1) \geq L \text{ and } \Pr_{x \sim D}(h^*(x) = 0) \geq L, \quad (3)$$

$$\Delta := |p^1 - p^0| \geq \epsilon/2. \quad (4)$$

We now outline a formal justification of Assumptions 3 and 4. For convenience, let $\Pr(h = \cdot)$ denote $\Pr_{x \sim D}(h(x) = \cdot)$ where it introduces little ambiguity.

Let p_h^1 (p_h^0) denote the expected payoff given that input x is labeled 1(0) by h. We have that p_h^1 has value

$$p^1 \Pr(h^* = 1 | h = 1) + p^0 \Pr(h^* = 0 | h = 1). \quad (5)$$

Now, suppose that $\Pr(h^* = 1) \leq L$ (the case of $\Pr(h^* = 0) \leq L$ follows by the same argument). Using the following theorem, it can shown that predictions made with respect to any $h \in H$ and sufficiently accurate estimates of p_h^1 and p_h^0 form a solution to APP.

Theorem 1 *For any positive real numbers ϵ and L, if $\Pr(h^* = 1) \leq L$, then for any $h \in H$, $\Pr\left(|p_h^{h(x)} - p^{h^*(x)}| > \epsilon\right) \leq 2L/\epsilon + L$.*

Next, suppose that $\Delta \leq \epsilon/2$. Since p_h^1 is a convex combination of p^1 and p^0 (see Equation 5), it lies between them. Hence, $|p_h^1 - p^1| \leq \epsilon/2$ and $|p_h^1 - p^0| \leq \epsilon/2$. The same holds for p_h^0. Thus, $\epsilon/2$-accurate estimates for p_h^1 and p_h^0 for any $h \in H$ suffice in this case (and can be obtained by the estimation method in Section 3.4).

3.4. Finite Hypothesis-Class Case

In this section, we demonstrate a solution to APP when the hypothesis class, H, is finite. From the reduction of Section 3.1, this case will produce a solution[2] to ABP (when H is finite). The computation of our method is proportional to $|H|$ (but has sample complexity proportional to $\lg |H|$) and, therefore, will only be practical for small hypothesis classes H.

Algorithm 1 SolveAPP

1: Draw m samples from EX. Let S be the multiset of labeled pairs drawn from EX.
2: For each $h_i \in H$ compute the following quantities:

$$\texttt{count}(h_i, 1) = |\{(x, y) \in S | h_i(x) = 1\}|$$

$$\texttt{count}(h_i, 0) = |\{(x, y) \in S | h_i(x) = 0\}|$$

$$\hat{p}^1_{h_i} = \frac{|\{(x, y) \in S | y = 1 \wedge h_i(x) = 1\}|}{\texttt{count}(h_i, 1)}$$

$$\hat{p}^0_{h_i} = \frac{|\{(x, y) \in S | y = 1 \wedge h_i(x) = 0\}|}{\texttt{count}(h_i, 0)}$$

3: Output some $h \in H$ and $\hat{p}^1_h$ and $\hat{p}^0_h$.

To fully specify our procedure, Algorithm 1, we need to provide a value for m in Step 1 and explain how to choose hypothesis h in Step 3. Now, $\hat{p}^1_{h_i}$ is the maximum likelihood estimate of $p^1_{h_i}$, the probability of receiving payoff of 1 when $h_i(x) = 1$. Depending on the number of samples x such that $h_i(x) = 1$ (that is, $\texttt{count}(h_i, 1)$), the estimate may be poor (when $\texttt{count}(h_i, 1)$ is small) or very accurate (when $\texttt{count}(h_i, 1)$ is large). Therefore, the values $\texttt{count}(h_i, 1)$ and $\texttt{count}(h_i, 0)$ are estimates of the accuracy of the estimates $\hat{p}^1_{h_i}$ and $\hat{p}^0_{h_i}$, respectively. Due to space limitations, we provide only a sketch of the correctness of Algorithm 1[3].

[2] It will only be an *efficient* solution if $\ln(|H|)$ is no larger than a polynomial of $\texttt{dim}(H)$.

[3] Full proofs will be made available in our technical report.

The following lemma is a direct consequence of the Hoeffding bound.

Lemma 1 *If hypothesis $h \in H$ satisfies $\Pr(h(x) = j) \geq L$ (for $j \in \{0, 1\}$) then after*

$$m_1(\alpha, L, \delta) = O\left((\alpha/L)\ln(\delta)\right)$$

labeled samples x (acquired from calls to $EX(h^*, p^1, p^0, D)$), $\texttt{count}(h, j) \geq \alpha$ holds, with probability at least $1 - \delta$.

We are now ready to specify the number of samples to draw. For $\delta' = \delta/(4|H|)$, let

$$m := m_1\left(\frac{\ln(1/\delta')}{2(\epsilon\mu/C_1)^2}, L, \delta'\right) = O\left(\frac{\ln(|H|/\delta)}{\epsilon^3\mu^3}\right) \quad (6)$$

for some constant C_1 that will be specified later. Let H' be the set of hypotheses h such that $\texttt{count}(h, 1) \geq \frac{\ln(1/\delta')}{2(\epsilon\mu/C_1)^2}$ and $\texttt{count}(h, 0) \geq \frac{\ln(1/\delta')}{2(\epsilon\mu/C_1)^2}$. By several applications of the Hoeffding bound, Lemma 1, and the union bound, it can be shown that, with probability at least $1 - \delta$, $h^* \in H'$ and for all $h \in H'$:

$$|\hat{p}^1_h - p^1_h| \leq \frac{\epsilon\mu}{C_1} \text{ and } |\hat{p}^0_h - p^0_h| \leq \frac{\epsilon\mu}{C_1}. \quad (7)$$

Thus, we assume these hold. Now, let

$$\bar{h} := \operatorname*{argmax}_{h \in H'} |\hat{p}^1_h - \hat{p}^0_h| \quad (8)$$

(breaking ties arbitrarily) be returned by our procedure (Step 3). We say that $\bar{h}$ is a hypothesis with *maximum empirical diversity*; according to our accurate but not exact estimates, the difference between the payoff rate when $\bar{h}(x) = 1$ rather than $\bar{h}(x) = 0$ is maximal. Thus, in some sense, $\bar{h}$ is our estimate of the most *informative* hypothesis. Next, we show that $\bar{h}$ suffices as a solution to APP.

We have that $|\hat{p}^1_{\bar{h}} - \hat{p}^0_{\bar{h}}| \geq |\hat{p}^1_{h^*} - \hat{p}^0_{h^*}| \geq |p^1_{h^*} - p^0_{h^*}| - 2(\epsilon\mu)/C_1 = |p^1 - p^0| - 2(\epsilon\mu)/C_1$. We conclude from this result and Equation 7 that the following hold:

$$\max\{p^1, p^0\} - \max\{p^1_{\bar{h}}, p^0_{\bar{h}}\} \leq \frac{3\epsilon\mu}{C_1}, \quad (9)$$

$$\min\{p^1_{\bar{h}}, p^0_{\bar{h}}\} - \min\{p^1, p^0\} \leq \frac{3\epsilon\mu}{C_1}. \quad (10)$$

To simplify the notation, define s and r, with $\{s, r\} = \{1, 0\}$, such that $p^s := \max\{p^1, p^0\}$, and $p^r := \min\{p^1, p^0\}$. Similarly, define $\bar{s}$ and $\bar{r}$, with $\{\bar{s}, \bar{r}\} = \{1, 0\}$, so $p^{\bar{s}} := \max\{p^1_{\bar{h}}, p^0_{\bar{h}}\}$ and $p^{\bar{r}} := \min\{p^1_{\bar{h}}, p^0_{\bar{h}}\}$.

Theorem 2 *Suppose $h^* \in H'$, Equation 7 holds, and let $\Delta := |p^1 - p^0| = p^s - p^r$. Then, we have that*

$$\Pr(h^*(x) = s | \bar{h}(x) = \bar{s}) \geq 1 - \frac{3\epsilon\mu}{C_1\Delta} \quad (11)$$

$$\textit{and} \quad \Pr(h^*(x) = r | \bar{h}(x) = \bar{r}) \geq 1 - \frac{3\epsilon\mu}{C_1\Delta}. \quad (12)$$

Proof sketch: We have that $0 \leq p^s - \bar{p}^s \leq 3\epsilon\mu/C_1$ and $0 \leq \bar{p}^i - p^i \leq 3\epsilon\mu/C_1$. Noting that $p^{\bar{s}}_{\bar{h}} = p^s \Pr(h^* = s|\bar{h} = \bar{s}) + p^r \Pr(h^* = r|\bar{h} = \bar{s})$, we have that $p^s - p^{\bar{s}}_{\bar{h}} = p^s - \Delta \Pr(h^* = s|\bar{h} = \bar{s}) - p^r = \Delta - \Delta \Pr(h^* = s|\bar{h} = \bar{s})$. In summary, $p^s - p^{\bar{s}}_{\bar{h}} = \Delta(1 - \Pr(h^* = s|\bar{h} = \bar{s})) \leq 3\epsilon\mu/C_1$. Rearranging yields Equation 11. The other cases follow by a similar argument. □

Let $C_1 = 12$. Note that $p^r \leq p^{\bar{r}}_{\bar{h}} \leq p^{\bar{s}}_{\bar{h}} \leq p^s$ always holds. We say that h^* and $\bar{h}$ *agree on input* x when either $h^*(x) = s$ and $\bar{h}(x) = \bar{s}$ both hold or $h^*(x) = r$ and $\bar{h}(x) = \bar{r}$ both hold. Using Theorem 2, it can be shown that the probability that h^* and $\bar{h}$ disagree is at most $(6\epsilon\mu)/(C_1\Delta)$. However, note that $(6\epsilon\mu)/(C_1\Delta) \leq (12\epsilon\mu)/(C_1\epsilon) = \mu$ (see Equation 4). When $h^*(x)$ and $\bar{h}(x)$ agree on an input x, we are guaranteed, by Equations 7, 9, and 10, that $|\hat{p}^{\bar{h}(x)}_{\bar{h}} - p^{h^*(x)}| \leq (4\epsilon\mu)/C_1 \leq \epsilon$, as desired.

To summarize, any instance of APP where H is finite can be solved with sample complexity as given in Equation 6.

4. Classification

The *classification* or *supervised learning* (SL) problem consists of an input space X, an unknown distribution $\mathcal{Y}$ over $X \times \{0, 1\}$, and a hypothesis space H. Let $EX := EX(\mathcal{Y})$ be an oracle that produces labeled pairs (x, y) distributed according to $\mathcal{Y}$. A learning sequence proceeds as follows. During each timestep, the algorithm calls EX to receive a labeled pair (x, y). After m calls, a hypothesis $h \in H$ is produced as output, chosen to minimize classification error,

$$\Pr_{(x,y)\sim\mathcal{Y}}(h(x) \neq y). \quad (13)$$

Let F be the function in H that minimizes this error.

Let $SL(X, H, \epsilon, \delta, EX)$ denote any *efficient* learning algorithm for the SL problem. It takes inputs $0 < \epsilon < 1$ and $0 < \delta < 1$. With probability at least $1 - \delta$, its output must satisfy:

$$\Pr_{(x,y)\sim\mathcal{Y}}(h(x) \neq y) \; - \Pr_{(x,y)\sim\mathcal{Y}}(F(x) \neq y) \leq \epsilon. \quad (14)$$

In other words, the error of the chosen hypothesis may not exceed that of the best hypothesis by more than ϵ. Furthermore, the number of calls to EX, denoted $SLSC(X, H, \epsilon, \delta, EX)$ is bounded by a polynomial in $1/\epsilon$, $1/\delta$, and $\mathtt{dim}(H)$.

4.1. Reduction to Associative Prediction

In this section, we show that a solution to the SL problem can be used to solve APP. We assume that H is closed under complementation[4]. For any Boolean statement S, let $[S]$ evaluate to 1 if S is true and 0 if S is false.

For any pair $y, y' \in \{0, 1\}$, consider the loss function:

$$l(y, y') := [y = 0 \wedge y' = 1]e_p + [y = 1 \wedge y' = 0]e_n \quad (15)$$

where e_p and e_n are real numbers satisfying $0 \leq e_p, e_n \leq 1$, and $e_p + e_n = 1$. We call e_p the *false-positive cost* and e_n the *false-negative cost*. Finding a minimizer of expected loss is the goal of *cost-sensitive classification* (Elkan, 2001) and can be reduced to the SL problem (Zadrozny et al., 2003). We exploit this reduction by providing costs, e_p and e_n, so that solving APP is achieved by viewing it as cost-sensitive classification.

We first observe that for a range of costs, the hypothesis that minimizes expected loss is h^* (or its complement). This connection is useful because APP predicts a (real-valued) probability, while cost-sensitive classification predicts a (Boolean-valued) label.

Lemma 2 *When $p^1 > p^0$, if $p^0/(1 - p^0) \leq e_p/e_n \leq p^1/(1 - p^1)$ then h^* has minimum expected loss. When $p^0 > p^1$, if $p^1/(1 - p^1) \leq e_p/e_n \leq p^0/(1 - p^0)$ then $\bar{h}^*$ (the complement of h^*) has minimum expected loss.*

Proof: For $h \in H$ and $x \in X$, the expected loss of h is $p^{h^*(x)}e_n$ when $h(x) = 0$ and $(1 - p^{h^*(x)})e_p$ when $h(x) = 1$. Thus, if $h^*(x) = 1$, then $h(x) = 1$ minimizes loss when $(1 - p^1)e_p \leq p^1 e_n$, which is equivalent to $e_p/e_n \leq (p^1)/(1 - p^1)$. If $h^*(x) = 0$, then $h(x) = 0$ minimizes loss when $e_p/e_n \geq (p^0)/(1 - p^0)$. Hence, if $p^1 > p^0$, h^* is the global minimizer when $p^0/(1 - p^0) \leq e_p/e_n \leq p^1/(1 - p^1)$. The second claim follows similarly. □

Since the case of $p^1 > p^0$ and $p^0 > p^1$ are symmetrical[5], from now on we assume that $p^1 > p^0$.

Lemma 2 suggests a choice for the error costs. Unfortunately, we cannot directly apply the bounds, as both p^1 and p^0 are unknown. Let Z denote the expected

[4] The complements can be added if necessary.

[5] For solving APP, $\bar{h}^*$ is as valid as h^*.

value of the input labels. The next lemma provides a simple formula, in terms of Z, for costs in the range given by Lemma 2. This formula is useful as accurate estimates of Z are easy to compute.

Let D be the distribution obtained from $\mathcal{Y}$ when restricted to X: $\Pr_{x\sim D}(x=a) := \Pr_{(x,y)\sim\mathcal{Y}}(x=a)$. We now revert to the short-hand notation of Section 3.3.

Lemma 3 *If* $Z = p^1 \Pr(h^*=1) + p^0 \Pr(h^*=0)$, *then* $Z/(1-Z)$ *is between* $p^0/(1-p^0)$ *and* $p^1/(1-p^1)$.

Proof: The result follows from viewing Z is a convex combination of p^1 and p^0. □

By Lemmas 2 and 3, if $e_p = Z$ and $e_n = 1 - Z$, then h^* is the minimizer of cost-based loss (Equation 15). Although Z is unknown, an accurate estimate can be obtained, with high probability, by sampling from EX. To ensure that near-optimal solutions to the SL problem correspond to near-optimal solutions to APP, we require costs well separated from either endpoint of the valid cost range (p^1 and p^0) and extreme values 0 and 1. Suppose we obtain an approximation $\hat{Z}$ of Z such that $|\hat{Z} - Z| \leq \epsilon/8$. Consider the set of costs:

$$\{e_{p_1}, e_{p_2}, e_{p_3}\} = \{\hat{Z} - \epsilon/4, \hat{Z}, \hat{Z} + \epsilon/4\}. \tag{16}$$

By Assumption 4, one of these quantities is at least $\epsilon/8$ away from either endpoint (p^0 and p^1) and between them. This property is not upset by rounding any value less than $\epsilon/8$ to $\epsilon/8$ and greater than $1-\epsilon/8$ to $1-\epsilon/8$.

We apply the technique of Zadrozny et al. (2003) by using a modified distribution[6] $\mathcal{Y}'$ (over pairs $(x,y) \in X \times \{0,1\}$) such that optimal classification on $\mathcal{Y}'$ is equal to optimal cost-sensitive classification on $\mathcal{Y}$. This new distribution is $\mathcal{Y}'(x,y) = \frac{c(y)\mathcal{Y}(x,y)}{E_{(x,y)\sim\mathcal{Y}}[c(y)]}$, where $c(0) = e_p$ and $c(1) = e_n$.

Given an oracle EX that samples according to $\mathcal{Y}$ we would like to simulate an oracle EX' that samples according to $\mathcal{Y}'$. We achieve this goal with *rejection sampling*[7] (Zadrozny et al., 2003). To simulate a call to EX', EX is called several times, each call producing a pair $(x,y) \in X \times \{0,1\}$. Let $e_{\max} = \max\{e_p, e_n\}$ and $e_{\min} = \min\{e_p, e_n\}$. If $y = 1$, then the sample is *accepted* with probability $e_n/e_{\max}$ (otherwise it is *rejected*). If $y = 0$, then the sample is *accepted* with probability $e_p/e_{\max}$. The output of EX' is the first pair that is accepted. Thus, the expected number of calls to EX for one call to EX' is at most $e_{\max}/e_{\min}$.

[6] Let $\mathcal{Y}(\cdot)$ denote the *probability density function* of $\mathcal{Y}$.

[7] This technique ensures that EX' produces independent samples given independent samples from EX.

Theorem 3 *For rejection-sampling-based oracle* EX', *if* $e_p \geq p^0 + \epsilon/8$, $e_p \leq p^1 - \epsilon/8$, *and* $\epsilon/8 \leq e_p, e_n \leq 1 - \epsilon/8$, *then algorithm* $SL(X, H, \epsilon^3\mu/(8C_1), \delta/C_2, EX')$ *outputs a hypothesis* h *such that* $\Pr_{(x,y)\sim\mathcal{Y}}(h(x) \neq h^*(x)) \leq \epsilon^2\mu/C_1$ *with probability* $1 - \delta/C_2$.

Proof sketch: The key to the proof is to break the error of h into two pieces: When $h^*(x) = 0$ and when $h^*(x) = 1$. The labels on each side are independently generated based on p^1 and p^0. Taking into account the modified distribution allows us to convert an error bound based on accuracy ($\Pr_{(x,y)\sim\mathcal{Y}'}(h(x) \neq y)$ from the SL algorithm to an error bound based on the distribution of instances ($\Pr_{(x,y)\sim\mathcal{Y}}(h^*(x) \neq h(x))$). This bound depends on the cost, e_p. The worst case is when e_p is close to either p^0 or p^1; however, we are able to bound the cost since e_p can only get within a distance of $\epsilon/8$ of either p^0 or p^1. □

Finally, we need to show that if a hypothesis agrees with h^* over most of the input space, then it solves APP (see Equation 2).

Theorem 4 *If* $\Pr(h \neq h^*) \leq U$, *then* $|p^1_h - p^1| \leq 2U/\Pr(h=1)$ *and* $|p^0_h - p^0| \leq 2U/\Pr(h=0)$.

Our reduction is as follows. First, we obtain $N_1 = O(\lg(1/\delta)/\epsilon^2)$ samples by calling EX (the oracle for APP). From these samples, we compute $\hat{Z} = \frac{\#\text{ samples labeled 1}}{N_1}$, which is used to define 3 separate oracles: for $i = 1,2,3$, Oracle EX_i is defined by costs e_{p_i} and $e_{n_i} := 1 - e_{p_i}$ via Equation 16 and can be simulated using EX as described above. Using an SL solver, we obtain three hypotheses, $h_i = SL(X, H, \epsilon^3\mu/(8C_1), \delta/C_2, EX_i)$, for $i = 1,2,3$ and constants $C_1 = 60$ and $C_2 = 5$.

By the above discussion, we are guaranteed that some hypothesis $h \in \{h_1, h_2, h_3\}$ satisfies $\Pr_D(h(x) \neq h^*(x)) \leq \epsilon^2\mu/C_1 \leq \mu$. Suppose we could obtain estimates $\hat{p}^1_h$ and $\hat{p}^0_h$ such that $|\hat{p}^1_h - p^1_h| \leq \epsilon/2$ and $|\hat{p}^0_h - p^0_h| \leq \epsilon/2$. Now, by Theorem 4 and Assumption 3, we have that $|\hat{p}^1_h - p^1| \leq |p^1_h - p^1| + \epsilon/2 \leq \frac{2\epsilon^2\mu}{C_1 \Pr(h=1)} + \frac{\epsilon}{2} \leq \frac{2\epsilon^2\mu}{C_1(L - \frac{\epsilon^2\mu}{C_1})} + \epsilon/2 \leq \frac{24\epsilon}{C_1 - 12} + \frac{\epsilon}{2} \leq \epsilon$. By the same argument we can show that $|\hat{p}^0_h - p^0| \leq \epsilon$. Hence, Equation 2 is satisfied.

Of course, we do not know which hypotheses in $\{h_1, h_2, h_3\}$ solve APP. We overcome this problem by treating $\{h_1, h_2, h_3\}$ as a new hypothesis space and using the techniques of Section 3.4. The formal analysis given in Section 3.4 assumed $h^* \in H$, which no longer holds. However, some $h \in \{h_1, h_2, h_3\}$ is an $\epsilon^2\mu/C_1$-approximation of h^*, with high probability. We don't have room for the details, but the method of Sec-

tion 3.4 is still valid because (a) Assumption 3 implies that $\Pr(h = 1) = \Omega(\mu\epsilon)$ and $\Pr(h = 0) = \Omega(\mu\epsilon)$, and (b) the diversity of h, $|p_h^1 - p_h^0|$, is $O(\epsilon)$ different from the diversity of h^*, $|p^1 - p^0|$. Our final reduction is outlined in pseudo code (Algorithm 2).

Algorithm 2 ReduceSL-APP

1: Draw N_1 samples from EX and compute $\hat{Z}$.
2: Create three different cost-sensitive classification problems with the three costs of Equation 16.
3: Reduce these three cost-sensitive problems to classification using the method of Zadrozny et al. (2003). Solve to yield three hypotheses: h_1, h_2, h_3.
4: Use the method of Section 3.4 to solve an instance of APP with the hypothesis set $\mathcal{H} = \{h_1, h_2, h_3\}$. This procedure results in a single hypothesis h and its associated estimates $\hat{p}_h^1$ and $\hat{p}_h^0$.
5: Output the prediction function $f(x) = \hat{p}^{h(x)}$.

Finally, we note that there are three possible sources of failure: estimating $\hat{Z}$, calling the SL learner (which is called 3 times), and applying the methods of Section 3.4. By setting $C_2 = 5$ and using the union bound, we can limit the total failure probability to at most $1 - \delta$. Since the sample complexity of the SL learner and the procedure of Section 3.4 are polynomial, the sample complexity of our reduction is also guaranteed to be polynomial in the relevant quantities.

The *Probing* algorithm of Langford and Zadrozny (2005) is very similar to our approach; however, it deals with a much more general problem. In their setting, the probability that an input x is labeled 1 can be any function of x. In our restricted problem (APP), it is one of only two possible values (p^1 or p^0) dependent on whether it is classified as 1 or 0 by a single hypothesis, h^*, which is itself restricted to some class, H, of Boolean-valued functions. This restriction allows for a simpler (and likely more efficient) solution and a different theoretical analysis. For instance, the Probing algorithm solves many different cost-sensitive classification problems, while our method performs only 3. The output (predictor) of Probing is a combination of all the learned classifiers, whereas in our scenario, only a single hypothesis h and two numbers $\hat{p}^1$ and $\hat{p}^0$ are needed to guarantee accurate predictions (predict $\hat{p}^h(x)$ on input x).

5. Solving the Classification Problem

Section 4.1 leaves open the question of whether the SL problem can be solved. By Theorem 5.1 of Kearns and Schapire (1990), if the VC dimension of H[8] is d then after $O((d/\epsilon^2)(\lg(1/\epsilon)+\lg(1/\delta)))$ labeled samples, the hypothesis $h \in H$ that minimizes empirical loss[9] has ϵ-optimal true loss, with probability at least $1 - \delta$. So, in terms of sample complexity, the problem is solvable. The computational complexity of computing the empirical loss minimizer (or a good approximation) depends on H.

6. Experiments

Kaelbling (1994) includes an empirical comparison of ABP learning algorithms on a set of "binomial Boolean expression worlds". In our terminology, this comparison focused on $X = \{0,1\}^n$ (n-bit input vectors), $k = 2$ (two arms), H equal to 2-DNF formulae, and D a uniform distribution. The experiments showed, in part, that generalizing algorithms outperformed non-generalizing algorithms when the input space was sufficiently large.

For our experiment, we examined a case with small H and large X. Let X be the set of all n-bit strings $x = (x_1, \ldots, x_n)$. The target hypothesis is $h^*(x) := x_1 \wedge \bar{x}_n$, and H is the set of all conjunctions of 2 literals. The problem has two arms (0 and 1) with payoff rates $p_0^1 = .5$ and $p_0^0 = .8$ for arm 0 and $p_1^1 = .9$ and $p_1^0 = .6$ for arm 1. The optimal policy is to choose arm 0 when $h^*(x) = 0$ and arm 1 otherwise. The expected reward of the optimal policy is $(.75)(.8) + (.25)(.9) = 0.825$.

We compared two algorithms. The first is the naïve algorithm, designed for the k-armed bandit problem. It chooses each arm m' times and finds the arm with highest sample mean payoff, which it then chooses forever (acting greedily). For large enough m', it will, with high probability, find an ϵ-optimal arm (Fong, 1995). The adaptation to ABP, called **Bandit-Naive**, is achieved by viewing each input $x \in X$ as a separate bandit problem; a valid, but often inefficient approach.

The second algorithm, called **ABP-Naive**, is that of Section 3.4. It has two parameters, m and θ. It first chooses each arm m times. Then, for each arm, it computes the hypothesis h with maximum empirical diversity (see Equation 8) with at least θ positive examples (x such that $h(x) = 1$) and at least θ negative examples (x such that $h(x) = 0$).

We varied the number of input bits from 2 to 10. Each algorithm was given 3000 input choices, and the average per-step reward is plotted, using the best parameter settings found by systematic search. Each experi-

[8] H must satisfy several "permissibility" assumptions.
[9] The one with the fewest mistakes on the training data.

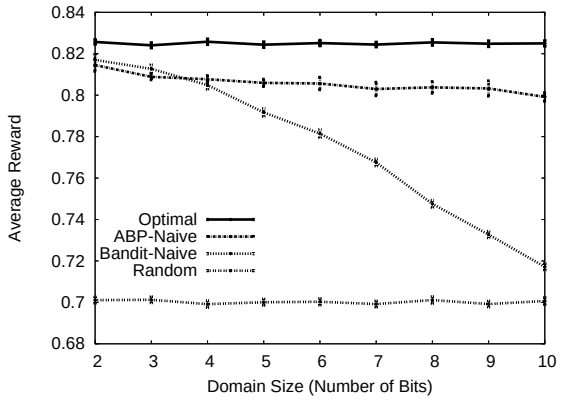

Figure 1. Comparing Bandit-Naive and ABP-Naive in a two-armed Associative Bandit Problem

ment was repeated 100 times and the results averaged.

As the problem size increases, we expect ABP-Naive, which generalizes over inputs, to outperform Bandit-Naive, which considers each input separately. When $n = 10$, for example, there are $2^{10} = 1024$ different inputs (different bandit problems), but only $2(10^2) - 2(10) = 180$ hypotheses. Our results confirm this hypothesis; see Figure 1.

7. Conclusion

We formalized the associative-bandit problem setting of Kaelbling (1994) as a computational learning theory problem. In this problem, inputs are selected from a fixed distribution. After each input is presented, the learner can choose from a fixed set of arms (actions) in an attempt to maximize its immediate expected reward. Each arm has its own payoff function, which is defined by an unknown hypothesis from a known hypothesis class along with separate payoff probabilities for inputs labeled 0 and 1 by this hypothesis.

Whereas earlier work took an empirical stance, we were able to prove formal bounds on the amount of experience trials needed to attain near-optimal utility with high probability. Our approach focused on generalization and used a primitive, but sufficient, exploration method. The formal proof showed that to learn to make optimal decisions in an associative bandit problem it is sufficient to make accurate predictions of the individual arms.

More generally, we showed how to reduce the associative prediction problem to cost-sensitive classification and then to standard classification. That is, we showed that a solution to the classification problem for a hypothesis class can be used to solve the corresponding associative-prediction problem.

Our next step is to apply the associative-bandit formalism to practical decision problems, such as choosing optimal networking parameters in a dynamic environment based on periodic measurements of network conditions and ultimately to the full reinforcement-learning setting.

Acknowledgments

Thanks to the National Science Foundation (IIS-0325281) and DARPA IPTO for support. We also thank John Langford, Rohan Fernandes, and our anonymous reviewers for suggestions.

References

Abe, N., Biermann, A. W., & Long, P. M. (2003). Reinforcement learning with immediate rewards and linear hypotheses. *Algorithmica*, *37*, 263–293.

Auer, P. (2000). An improved on-line algorithm for learning linear evaluation functions. *Proceedings of the 13th Annual Conference on Computational Learning Theory* (pp. 118–125).

Berry, D. A., & Fristedt, B. (1985). *Bandit problems: Sequential allocation of experiments*. London, UK: Chapman and Hall.

Elkan, C. (2001). The foundations of cost-sensitive learning. *IJCAI* (pp. 973–978).

Fiechter, C.-N. (1995). PAC associative reinforcement learning. Unpublished manuscript.

Fiechter, C.-N. (1997). Expected mistake bound model for on-line reinforcement learning. *Proceedings of the Fourteenth International Conference on Machine Learning* (pp. 116–124).

Fong, P. W. L. (1995). A quantitative study of hypothesis selection. *Proceedings of the Twelfth International Conference on Machine Learning (ICML-95)* (pp. 226–234).

Kaelbling, L. P. (1994). Associative reinforcement learning: Functions in k-DNF. *Machine Learning*, *15*.

Kearns, M. J., & Schapire, R. E. (1990). Efficient distribution-free learning of probabilistic concepts. *Journal of Computer and System Sciences*, *48*, 464–497.

Langford, J., & Zadrozny, B. (2005). Estimating class membership probabilities using classifier learners. *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics* (pp. 198–205).

Mitchell, T. M. (1997). *Machine learning*. McGraw Hill.

Valiant, L. G. (1984). A theory of the learnable. *Communications of the ACM*, *27*, 1134–1142.

Zadrozny, B., Langford, J., & Abe, N. (2003). Cost-sensitive learning by cost-proportionate example weighting. *ICDM* (p. 435).