Contextual Bandit Algorithms with Supervised Learning Guarantees

Alina Beygelzimer
IBM Research
Hawthorne, NY
beygel@us.ibm.com

John Langford
Yahoo! Research
New York, NY
jl@yahoo-inc.com

Lihong Li
Yahoo! Research
Santa Clara, CA
lihong@yahoo-inc.com

Lev Reyzin
Georgia Institute of Technology
Atlanta, GA
lreyzin@cc.gatech.edu

Robert E. Schapire
Princeton University
Princeton, NJ
schapire@cs.princeton.edu

Abstract

We address the problem of competing with any large set of N policies in the non-stochastic bandit setting, where the learner must repeatedly select among K actions but observes only the reward of the chosen action.

We present a modification of the `Exp4` algorithm of Auer et al. [2], called `Exp4.P`, which with high probability incurs regret at most $O(\sqrt{KT \ln N})$. Such a bound does not hold for `Exp4` due to the large variance of the importance-weighted estimates used in the algorithm. The new algorithm is tested empirically in a large-scale, real-world dataset. For the stochastic version of the problem, we can use `Exp4.P` as a subroutine to compete with a possibly infinite set of policies of VC-dimension d while incurring regret at most $O(\sqrt{Td \ln T})$ with high probability.

These guarantees improve on those of all previous algorithms, whether in a stochastic or adversarial environment, and bring us closer to providing guarantees for this setting that are comparable to those in standard supervised learning.

1 INTRODUCTION

A learning algorithm is often faced with the problem of acting given feedback only about the actions that it has taken in the past, requiring the algorithm to explore. A canonical example is the problem of personalized content recommendation on web portals, where the goal is to learn which items are of greatest interest to a user, given such observable context as the user's search queries or geolocation.

Formally, we consider an online bandit setting where at every step, the learner observes some contextual information and must choose one of K actions, each with a potentially different reward on every round. After the decision is made, the reward of the chosen action is revealed. The learner has access to a class of N policies, each of which also maps context to actions; the learner's performance is measured in terms of its *regret* to this class, defined as the difference between the cumulative reward of the best policy in the class and the learner's reward.

This setting goes under different names, including the "partial-label problem" [11], the "associative bandit problem" [18], the "contextual bandit problem" [13] (which is the name we use here), the "k-armed (or multi-armed) bandit problem with expert advice" [2], and "associative reinforcement learning" [9]. Policies are sometimes referred to as hypotheses or experts, and actions are referred to as arms.

If the total number of steps T (usually much larger than K) is known in advance, and the contexts and rewards are sampled independently from a fixed but unknown joint distribution, a simple solution is to first choose actions uniformly at random for $O(T^{2/3})$ rounds, and from that point on use the policy that per-

Appearing in Proceedings of the 14^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2011, Fort Lauderdale, FL, USA. Volume 15 of JMLR: W&CP 15. Copyright 2011 by the authors.

formed best on these rounds. This approach, a variant of ϵ-greedy (see [19]), sometimes called ϵ-first, can be shown to have a regret bound of $O\left(T^{2/3}(K \ln N)^{1/3}\right)$ with high probability [13]. In the full-label setting, where the entire reward vector is revealed to the learner at the end of each step, the standard machinery of supervised learning gives a regret bound of $O(\sqrt{T \ln N})$ with high probability, using the algorithm that predicts according to the policy with the currently lowest empirical error rate.

This paper presents the first algorithm, `Exp4.P`, that with high probability achieves $O(\sqrt{TK \ln N})$ regret in the adversarial contextual bandit setting. This improves on the $O(T^{2/3}(K \ln N)^{1/3})$ high probability bound in the stochastic setting. Previously, this result was known to hold *in expectation* for the algorithm `Exp4` [2], but a high probability statement did not hold for the same algorithm, as per-round regrets on the order of $O(T^{-1/4})$ were possible [2]. Succeeding with high probability is important because reliably useful methods are preferred in practice.

The `Exp4.P` analysis addresses competing with a finite (but possibly exponential in T) set of policies. In the stochastic case, ϵ-greedy or epoch-greedy style algorithms [13] can compete with an infinite set of policies with a finite VC-dimension, but the worst-case regret grows as $O(T^{2/3})$ rather than $O(T^{1/2})$. We show how to use `Exp4.P` in a black-box fashion to guarantee a high probability regret bound of $O(\sqrt{Td \ln T})$ in this case, where d is the VC-dimension. There are simple examples showing that it is impossible to compete with a VC-set with an online adaptive adversary, so some stochastic assumption seems necessary here.

This paper advances a basic argument, namely, that such exploration problems are solvable in almost exactly the same sense as supervised learning problems, with suitable modifications to existing learning algorithms. In particular, we show that learning to compete with any set of strategies in the contextual bandit setting requires only a factor of K more experience than for supervised learning (to achieve the same level of accuracy with the same confidence).

`Exp4.P` does retain one limitation of its predecessors—it requires keeping explicit weights over the experts, so in the case when N is too large, the algorithm becomes inefficient. On the other hand, `Exp4.P` provides a practical framework for incorporating more expressive expert classes, and it is efficient when N is polynomial in K and T. It may also be possible to run `Exp4.P` efficiently in certain cases when working with a family of experts that is exponentially large, but well structured, as in the case of experts corresponding to all prunings of a decision tree [8]. A concrete example of this approach is given in Section 7, where an efficient implementation of `Exp4.P` is applied to a large-scale real-world problem.

Related work: The non-contextual K-armed bandit problem was introduced by Robbins [17], and analyzed by Lai and Robbins [12] in the i.i.d. case for fixed reward distributions.

An adversarial version of the bandit problem was introduced by Auer et al. [2]. They gave an exponential-weight algorithm called `Exp3` with expected cumulative regret of $\tilde{O}(\sqrt{KT})$ and also `Exp3.P` with a similar bound that holds with high probability. They also showed that these are essentially optimal by proving a matching lower bound, which holds even in the i.i.d. case. They were also the first to consider the K-armed bandit problem with expert advice, introducing the `Exp4` algorithm as discussed earlier. Later, McMahan and Streeter [16] designed a cleaner algorithm that improves on their bounds when many irrelevant actions (that no expert recommends) exist. Further background on online bandit problems appears in [5].

`Exp4.P` is based on a careful composition of the `Exp4` and `Exp3.P` algorithms. We distill out the exact exponential moment method bound used in these results, proving an inequality for martingales (Theorem 1) to derive a sharper bound more directly. Our bound is a Freedman-style inequality for martingales [6], and a similar approach was taken in Lemma 2 of Bartlett et al. [3]. Our bound, however, is more elemental than Bartlett et al.'s since our Theorem can be used to prove (and even tighten) their Lemma, but not vice versa.

With respect to competing with a VC-set, a claim similar to our Theorem 5 (Section 5) appears in a work of Lazaric and Munos [14]. Although they incorrectly claimed that `Exp4` can be analyzed to give a regret bound of $\tilde{O}(KT \ln N)$ with high probability, one can use `Exp4.P` in their proof instead. Besides being correct, our analysis is tighter, which is important in many situations where such a risk-sensitive algorithm might be applied.

Related to the bounded VC-dimension setting, Kakade and Kalai [10] give a $O(T^{3/4})$ regret guarantee for the transductive online setting, where the learner can observe the rewards of all actions, not only those it has taken. In [4], Ben-David et al. consider agnostic online learning for bounded Littlestone-dimension. However, as VC-dimension does not bound Littlestone dimension, our work provides much tighter bounds in many cases.

Possible Approaches for a High Probability Algorithm

To develop a better intuition about the problem, we describe several naive strategies and illustrate why they fail. These strategies fail even if the rewards of each arm are drawn independently from a fixed unknown distribution, and thus certainly fail in the adversarial setting.

Strategy 1: Use confidence bounds to maintain a set of plausible experts, and randomize uniformly over the actions predicted by at least one expert in this set. To see how this strategy fails, consider two arms, 1 and 0, with respective deterministic rewards 1 and 0. The expert set contains N experts. At every round, one of them is chosen uniformly at random to predict arm 0, and the remaining $N-1$ predict arm 1. All of the experts have small regret with high probability. The strategy will randomize uniformly over both arms on every round, incurring expected regret of nearly $T/2$.

Strategy 2: Use confidence bounds to maintain a set of plausible experts, and follow the prediction of an expert chosen uniformly at random from this set. To see how this strategy fails, let the set consist of $N > 2T$ experts predicting in some set of arms, all with reward 0 at every round, and let there be a good expert choosing another arm, which always has reward 1. The probability we never choose the good arm is $(1-1/N)^T$. We have $-T\log(1-\frac{1}{N}) < T\frac{\frac{1}{N}}{1-\frac{1}{N}} \leq \frac{2T}{N} < 1$, using the elementary inequality $-\log(1-x) < x/(1-x)$ for $x \in (0,1]$. Thus $(1-1/N)^T > \frac{1}{2}$, and the strategy incurs regret of T with probability greater than 1/2 (as it only observes 0 rewards and is unable to eliminate any of the bad experts).

2 PROBLEM SETTING AND NOTATION

Let $\boldsymbol{r}(t) \in [0,1]^K$ be the vector of rewards, where $r_j(t)$ is the reward of arm j on round t. Let $\boldsymbol{\xi}^i(t)$ be the K-dimensional advice vector of expert i on round t. This vector represents a probability distribution over the arms, in which each entry $\xi^i_j(t)$ is the (expert's recommendation for the) probability of choosing arm j. For readability, we always use $i \in \{1,\ldots,N\}$ to index experts and $j \in \{1,\ldots,K\}$ to index arms.

For each policy π, the associated expert predicts according to $\pi(x_t)$, where x_t is the context available in round t. As the context is only used in this fashion here, we talk about expert predictions as described above. For a deterministic π, the corresponding prediction vector has a 1 in component $\pi(x_t)$ and 0 in the remaining components.

On each round t, the world commits to $\boldsymbol{r}(t) \in [0,1]^K$. Then the N experts make their recommendations $\boldsymbol{\xi}^1(t),\ldots,\boldsymbol{\xi}^N(t)$, and the learning algorithm A (seeing the recommendations but not the rewards) chooses action $j_t \in \{1,\ldots,K\}$. Finally, the world reveals reward $r_{j_t}(t)$ to the learner, and this game proceeds to the next round.

We define the *return* (cumulative reward) of A as $G_A \doteq \sum_{t=1}^T r_{j_t}(t)$. Letting $y_i(t) = \boldsymbol{\xi}^i(t) \cdot \boldsymbol{r}(t)$, we also define the expected return of expert i,

$$G_i \doteq \sum_{t=1}^T y_i(t),$$

and $G_{\max} = \max_i G_i$. The expected *regret* of algorithm A is defined as

$$G_{\max} - \mathbf{E}[G_A].$$

We can also think about bounds on the regret which hold with arbitrarily high probability. In that case, we can say that the regret is bounded by ϵ with probability $1-\delta$, if we have

$$\mathbf{Pr}[G_{\max} - G_A > \epsilon] \leq \delta.$$

In the definitions of expected regret and the high probability bound, the probabilities and expectations are taken w.r.t. both the randomness in the rewards $\boldsymbol{r}(t)$ and the algorithm's random choices.

3 A GENERAL RESULT FOR MARTINGALES

Before proving our main result (Theorem 2), we prove a general result for martingales in which the variance is treated as a random variable. It is used in the proof of Lemma 3 and may also be of independent interest. The technique is the standard one used to prove Bernstein's inequality for martingales [6]. The useful difference here is that we prove the bound for any fixed *estimate* of the variance rather than any *bound* on the variance.

Let $X_1,\ldots,X_T$ be a sequence of real-valued random variables. Let $\mathbf{E}_t[Y] = \mathbf{E}[Y|X_1,\ldots,X_{t-1}]$.

Theorem 1. *Assume, for all t, that $X_t \leq R$ and that $\mathbf{E}_t[X_t] = 0$. Define the random variables*

$$S \doteq \sum_{t=1}^T X_t, \qquad V \doteq \sum_{t=1}^T \mathbf{E}_t\left[X_t^2\right].$$

Then for any $\delta > 0$, with probability at least $1-\delta$, we have the following guarantee:

For any $V' \in \left[\frac{R^2\ln(1/\delta)}{e-2}, \infty\right)$,

$$S \leq \sqrt{(e-2)\ln(1/\delta)}\left(\frac{V}{\sqrt{V'}} + \sqrt{V'}\right)$$

and for $V' \in \left[0, \frac{R^2 \ln(1/\delta)}{e-2}\right]$,

$$S \leq R \ln(1/\delta) + (e-2)\frac{V}{R}.$$

Note that a simple corollary of this theorem is the more typical Freedman-style inequality, which depends on an *a priori* upper bound, which can be substituted for V' and V.

Proof. For a fixed $\lambda \in [0, 1/R]$, conditioning on $X_1, \ldots, X_{t-1}$ and computing expectations gives

$$\begin{aligned} \mathbf{E}_t\left[e^{\lambda X_t}\right] &\leq \mathbf{E}_t\left[1 + \lambda X_t + (e-2)\lambda^2 X_t^2\right] && (1)\\ &= 1 + (e-2)\lambda^2 \mathbf{E}_t\left[X_t^2\right] && (2)\\ &\leq \exp\left((e-2)\lambda^2 \mathbf{E}_t\left[X_t^2\right]\right). && (3) \end{aligned}$$

Eq. (1) uses the fact that $e^z \leq 1 + z + (e-2)z^2$ for $z \leq 1$. Eq. (2) uses $\mathbf{E}_t[X_t] = 0$. Eq. (3) uses $1+z \leq e^z$ for all z.

Let us define random variables $Z_0 = 1$ and, for $t \geq 1$,

$$Z_t = Z_{t-1} \cdot \exp\left(\lambda X_t - (e-2)\lambda^2 \mathbf{E}_t\left[X_t^2\right]\right).$$

Then,

$$\begin{aligned} \mathbf{E}_t[Z_t] &= Z_{t-1} \cdot \exp\left(-(e-2)\lambda^2 \mathbf{E}_t\left[X_t^2\right]\right) \cdot \mathbf{E}_t\left[e^{\lambda X_t}\right]\\ &\leq Z_{t-1} \cdot \exp\left(-(e-2)\lambda^2 \mathbf{E}_t\left[X_t^2\right]\right)\\ &\quad \cdot \exp\left((e-2)\lambda^2 \mathbf{E}_t\left[X_t^2\right]\right) = Z_{t-1}. \end{aligned}$$

Therefore, taking expectation over all of the variables $X_1, \ldots, X_T$ gives

$$\mathbf{E}[Z_T] \leq \mathbf{E}[Z_{T-1}] \leq \cdots \leq \mathbf{E}[Z_0] = 1.$$

By Markov's inequality, $\mathbf{Pr}[Z_T \geq 1/\delta] \leq \delta$. Since

$$Z_T = \exp\left(\lambda S - (e-2)\lambda^2 V\right),$$

we can substitute $\lambda = \min\left\{\frac{1}{R}, \sqrt{\frac{\ln(1/\delta)}{(e-2)V'}}\right\}$ and apply algebra to prove the theorem. □

4 A HIGH PROBABILITY ALGORITHM

The `Exp4.P` algorithm is given in Algorithm 1. It comes with the following guarantee.

Theorem 2. *Assume that* $\ln(N/\delta) \leq KT$, *and that the set of experts includes one which, on each round, selects an action uniformly at random. Then, with probability at least* $1-\delta$,

$$G_{\text{Exp4.P}} \geq G_{\max} - 6\sqrt{KT \ln(N/\delta)}.$$

Algorithm 1 Exp4.P

```
parameters: $\delta > 0$, $p_{\min} \in [0, 1/K]$
$\left(\text{we set } p_{\min} = \sqrt{\frac{\ln N}{KT}}\right)$
initialization:  Set $w_i(1) = 1$ for $i = 1, \ldots, N$.
for each $t = 1, 2, \ldots$
    1. get advice vectors $\boldsymbol{\xi}^1(t), \ldots, \boldsymbol{\xi}^N(t)$
    2. set $W_t = \sum_{i=1}^N w_i(t)$ and for $j = 1, \ldots, K$ set
           $p_j(t) = (1 - Kp_{\min}) \sum_{i=1}^N \frac{w_i(t)\xi_j^i(t)}{W_t} + p_{\min}$
    3. draw action $j_t$ randomly according to the probabilities $p_1(t), \ldots, p_K(t)$.
    4. receive reward $r_{j_t}(t) \in [0, 1]$.
    5. for $j = 1, \ldots, K$ set
           $\hat{r}_j(t) = \begin{cases} r_j(t)/p_j(t) & \text{if } j = j_t \\ 0 & \text{otherwise} \end{cases}$
    6. for $i = 1, \ldots, N$ set
           $\hat{y}_i(t) = \boldsymbol{\xi}^i(t) \cdot \hat{\boldsymbol{r}}(t)$
           $\hat{v}_i(t) = \sum_j \xi_j^i(t)/p_j(t)$
           $w_i(t+1) = w_i(t) e^{\left(\frac{p_{\min}}{2}\left(\hat{y}_i(t) + \hat{v}_i(t)\sqrt{\frac{\ln(N/\delta)}{KT}}\right)\right)}$
```

The proof of this theorem relies on two lemmas. The first lemma gives an upper confidence bound on the expected reward of an expert given the estimated reward of that expert.

The estimated reward of an expert is defined as

$$\hat{G}_i \doteq \sum_{t=1}^T \hat{y}_i(t).$$

We also define

$$\hat{\sigma}_i \doteq \sqrt{KT} + \frac{1}{\sqrt{KT}} \sum_{t=1}^T \hat{v}_i(t).$$

Lemma 3. *Under the conditions of Theorem 2,*

$$\mathbf{Pr}\left[\exists i : G_i \geq \hat{G}_i + \sqrt{\ln(N/\delta)}\hat{\sigma}_i\right] \leq \delta.$$

Proof. Fix i. Recalling that $y_i(t) = \boldsymbol{\xi}^i(t) \cdot \boldsymbol{r}(t)$ and the definition of $\hat{y}_i(t)$ in Algorithm 1, let us further define the random variables $X_t = y_i(t) - \hat{y}_i(t)$ to which we will apply Theorem 1. Then $\mathbf{E}_t\left[\hat{y}_i(t)\right] = y_i(t)$ so that $\mathbf{E}_t\left[X_t\right] = 0$ and $X_t \leq 1$. Further, we can compute

$$\begin{aligned}
\mathbf{E}_t\left[X_t^2\right] &= \mathbf{E}_t\left[(y_i(t) - \hat{y}_i(t))^2\right] \\
&= \mathbf{E}_t\left[\hat{y}_i(t)^2\right] - y_i(t)^2 \leq \mathbf{E}_t[\hat{y}_i(t)^2] \\
&= \mathbf{E}_t\left[\left(\boldsymbol{\xi}^i(t) \cdot \hat{\boldsymbol{r}}(t)\right)^2\right] \\
&= \sum_j p_j(t)\left(\xi_j^i(t) \cdot \frac{r_j(t)}{p_j(t)}\right)^2 \\
&\leq \sum_j \frac{\xi_j^i(t)}{p_j(t)} \\
&= \hat{v}_i(t).
\end{aligned}$$

Note that

$$G_i - \hat{G}_i = \sum_{t=1}^{T} X_t.$$

Using δ/N instead of δ, and setting $V' = KT$ in Theorem 1 gives us

$$\mathbf{Pr}\left[G_i - \hat{G}_i \geq \sqrt{(e-2)\ln\left(\tfrac{N}{\delta}\right)}\left(\tfrac{\sum_{t=1}^{T}\hat{v}_i(t)}{\sqrt{KT}} + \sqrt{KT}\right)\right] \\ \leq \\ \delta/N$$

Noting that $e-2 < 1$, and applying a union bound over the N experts gives the statement of the lemma. □

To state the next lemma, define

$$\hat{U} = \max_i \left(\hat{G}_i + \hat{\sigma}_i \cdot \sqrt{\ln(N/\delta)}\right).$$

Lemma 4. *Under the conditions of Theorem 2,*

$$\begin{aligned}
G_{\text{Exp4.P}} \geq & \left(1 - 2\sqrt{\frac{K \ln N}{T}}\right)\hat{U} - 2\sqrt{KT\ln(N/\delta)} \\
& -\sqrt{KT\ln N} - \ln(N/\delta).
\end{aligned}$$

We can now prove Theorem 2.

Proof. Taking the statement of Lemma 4 and applying the result of Lemma 3, and we get, with probability at least $1-\delta$,

$$\begin{aligned}
G_{\text{Exp4.P}} &\geq G_{\max} - 2\sqrt{\frac{K\ln N}{T}}T - \ln(N/\delta) \quad (4) \\
&\quad -\sqrt{KT\ln N} - 2\sqrt{KT\ln(N/\delta)} \\
&\geq G_{\max} - 6\sqrt{KT\ln(N/\delta)},
\end{aligned}$$

with Eq. (4) using $G_{\max} \leq T$. □

5 COMPETING WITH SETS OF FINITE VC DIMENSION

A standard VC-argument in the online setting can be used to apply `Exp4.P` to compete with an infinite set of policies Π with a finite VC dimension d, when the data is drawn independently from a fixed, unknown distribution. For simplicity, this section assumes that there are only two actions ($K=2$), as that is standard for the definition of VC-dimension.

The algorithm `VE` chooses an action uniformly at random for the first $\tau = \sqrt{T(2d\ln\frac{eT}{d} + \ln\frac{2}{\delta})}$ rounds. This step partitions Π into equivalence classes according to the sequence of advice on the first τ rounds. The algorithm constructs a finite set of policies Π' by taking one (arbitrary) policy from each equivalence class, and runs `Exp4.P` for the remaining $T-\tau$ steps using Π' as its set of experts.

For a set of policies Π, define $G_{\max(\Pi)}$ as the return of the best policy in Π at time horizon T.

Theorem 5. *For all distributions D over contexts and rewards, for all sets of policies Π with VC dimension d, with probability $1-\delta$,*

$$. \quad G_{VE} \geq G_{\max(\Pi)} - 9\sqrt{2T\left(d\ln\frac{eT}{d} + \ln\frac{2}{\delta}\right)}.$$

Proof. The regret of the initial exploration is bounded by τ. We first bound the regret of `Exp4.P` to Π', and the regret of Π' to Π. We then optimize with respect to τ to get the result.

Sauer's lemma implies that $|\Pi'| \leq \left(\frac{e\tau}{d}\right)^d$ and hence with probability $1-\delta/2$, we can bound $G_{\text{Exp4.P}}(\Pi', T-\tau)$ from below by

$$G_{\max(\Pi')} - 6\sqrt{2(T-\tau)(d\ln(e\tau/d) + \ln(2/\delta))}.$$

To bound the regret of Π' to Π, pick any sequence of feature observations $x_1, ..., x_T$. Sauer's Lemma implies the number of unique functions on the observation sequence in Π is bounded by $\left(\frac{eT}{d}\right)^d$.

For a uniformly random subset S of size τ of the feature observations we bound the probability that two functions π, π' agree on the subset. Let $n = n(\pi, \pi')$ be the number of disagreements on the T-length sequence. Then

$$\mathbf{Pr}_S\left[\forall x \in S \;\; \pi(x) = \pi'(x)\right] = \left(1 - \frac{n}{T}\right)^\tau \leq e^{-\frac{n\tau}{T}}.$$

Thus for all $\pi, \pi' \in \Pi$ with $n(\pi,\pi') \geq \frac{T}{\tau}\ln 1/\delta_0$, we have $\mathbf{Pr}_S\left[\forall x \in S \;\; \pi(x) = \pi'(x)\right] \leq \delta_0$.

Setting $\delta_0 = \frac{\delta}{2}\left(\frac{d}{eT}\right)^{2d}$ and using a union bound over every pair of policies, we get

$$\begin{aligned}\mathbf{Pr}_S(\exists \pi, \pi' \quad & \text{s.t. } n(\pi,\pi') \geq \tfrac{T}{\tau}\left(2d \ln \tfrac{eT}{d} + \ln \tfrac{2}{\delta}\right) \\ & \text{s.t. } \forall x \in S \;\; \pi(x) = \pi'(x)) \leq \delta/2.\end{aligned}$$

In other words, for all sequences $x_1, ..., x_T$ with probability $1-\delta/2$ over a random subset of size τ

$$G_{\max(\Pi')} \geq G_{\max(\Pi)} - \frac{T}{\tau}\left(2d \ln \frac{eT}{d} + \ln \frac{2}{\delta}\right).$$

Because the above holds for any sequence $x_1, ..., x_T$, it holds in expectation over sequences drawn i.i.d. from D. Furthermore, we can regard the first τ samples as the random draw of the subset since i.i.d. distributions are exchangeable.

Consequently, with probability $1-\delta$, we have

$$\begin{aligned} G_{\mathrm{VE}} \quad \geq \quad & G_{\max(\Pi)} - \tau - \frac{T}{\tau}\left(2d \ln \frac{eT}{d} + \ln \frac{2}{\delta}\right) \\ & -6\sqrt{2T(d\ln(e\tau/d) + \ln(2/\delta))}. \end{aligned}$$

Letting $\tau = \sqrt{T(2d \ln \frac{eT}{d} + \ln \frac{2}{\delta})}$ and substituting $T \geq \tau$ we get

$$G_{\mathrm{VE}} \geq G_{\max(\Pi)} - 9\sqrt{2T(d \ln \frac{eT}{d} + \ln \frac{2}{\delta})}.$$

□

This theorem easily extends to more than two actions ($K > 2$) given generalizations of the VC-dimension to multiclass classification and of Sauer's lemma [7].

6 A PRACTICAL IMPROVEMENT TO EXP4.P

Here we give a variant of Step 2 of Algorithm 1 for setting the probabilities $p_j(t)$, in the style of [16]. For our analysis of `Exp4.P`, the two properties we need to ensure in setting the probabilities $p_j(t)$ are

1. $p_j(t) \approx \sum_{i=1}^{N} \frac{w_i(t)\xi_j^i(t)}{W_t}$.
2. The value of each $p_j(t)$ is at least $p_{\min}$.

One way to achieve this, as is done in Algorithm 1, is to mix in the uniform distribution over all arms. While this yields a simpler algorithm and achieves optimal regret up to a multiplicative constant, in general, this technique can add unnecessary probability mass to badly-performing arms; for example it can double the probability of arms whose probability would already be set to $p_{\min}$.

Algorithm 2 An Alternate Method for Setting Probabilities in Step 2 of Algorithm 1

```
parameters: $w_1(t), w_2(t), \ldots w_N(t)$ and $\boldsymbol{\xi}^1(t), \ldots, \boldsymbol{\xi}^N(t)$ and $p_{\min}$

set
    $W_t = \sum_{i=1}^{N} w_i(t)$

for $j = 1$ to $K$ set
    $p_j(t) = \sum_{i=1}^{N} \frac{w_i(t)\xi_j^i(t)}{W_t}$

let $\Delta := 0$ and $l := 1$

for each action $j$ in increasing order according to $p_j$
    1. if $p_j\,(1 - \Delta/l) \geq p_{\min}$
       for all actions $j'$ with $p_{j'} \geq p_j$
       $p'_{j'} = p_{j'}\,(1 - \Delta/l)$
       return $\forall j \;\; p'_j$
    2. else $p'_j = p_{\min}, \;\; \Delta := \Delta + p'_j - p_j, \;\; l := l - p_j$.
```

A fix to this, first suggested by [16], is to ensure the two requirements via a different mechanism. We present a variant of their suggestion in Algorithm 2, which can be used to make `Exp4.P` perform better in practice with a computational complexity of $O(K \ln K)$ for computing the probabilities $p_j(t)$ per round. The basic intuition of this algorithm is that it enforces the minimum probability in order from smallest to largest action probability, while otherwise minimizing the ratio of the initial to final action probability.

This technique ensures our needed properties, and it is easy to verify that by setting probabilities using Algorithm 2 the proof in Section 4 remains valid with little modification. We use this variant in the experiments in Section 7.

7 EXPERIMENTS

In this section, we applied `Exp4.P` with the improvement in Section 6 to a large-scale contextual bandit problem. The purpose of the experiments is two-fold: it gives a proof-of-concept demonstration of the performance of `Exp4.P` in a non-trivial problem, and also illustrates how the algorithm may be implemented efficiently for special classes of experts.

The problem we study is personalized news article recommendation on the Yahoo! front page [1, 15]. Each time a user visits the front page, a news article out of a small pool of hand-picked candidates is highlighted. The goal is to highlight the most interesting articles to users, or formally, maximize the total number of user clicks on the recommended articles. In this problem, we treat articles as arms, and define the payoff to be 1 if the article is clicked on and 0 otherwise. Therefore, the average per-trial payoff of an algorithm/policy is the overall **click-through rate** (or **CTR** for short).

Following [15], we created $B = 5$ user clusters and thus each user, based on *normalized* Euclidean distance to the cluster centers, was associated with a B-dimensional *membership feature* $\mathbf{d}$ whose (nonnegative) components always sum up to 1. Experts are designed as follows. Each expert is associated with a mapping from user clusters to articles, that is, with a vector $\mathbf{a} \in \{1, \ldots, K\}^B$ where a_b is the article to be displayed for users from cluster $b \in \{1, \ldots, B\}$. When a user arrives with feature $\mathbf{d}$, the prediction $\boldsymbol{\xi}^{\mathbf{a}}$ of expert $\mathbf{a}$ is $\xi^{\mathbf{a}}_j = \sum_{b:a_b=j} d_b$. There are a total of K^B experts.

Now we show how to implement `Exp4.P` efficiently. Referring to the notation in `Exp4.P`, we have

$$\begin{aligned}\hat{y}_{\mathbf{a}}(t) &= \boldsymbol{\xi}^{\mathbf{a}}(t) \cdot \hat{\boldsymbol{r}}(t) = \sum_j \sum_{b:a_b=j} d_b(t)\hat{r}_j(t) \\ &= \sum_b d_b(t)\hat{r}_{a_b}(t), \\ \hat{v}_{\mathbf{a}}(t) &= \sum_j \sum_{b:a_b=j} \frac{d_b(t)}{p_j(t)} = \sum_b \frac{d_b(t)}{p_{a_b}(t)}.\end{aligned}$$

Thus,

$$\begin{aligned}&w_{\mathbf{a}}(t+1) \\ &\quad = w_{\mathbf{a}}(t)\exp\left(\frac{p_{\min}}{2}\left(\hat{y}_{\mathbf{a}}(t) + \hat{v}_{\mathbf{a}}(t)\sqrt{\frac{\ln(N/\delta)}{KT}}\right)\right) \\ &\quad = w_{\mathbf{a}}(t)\exp\left(\sum_b d_b(t) f_{a_b}(t)\right),\end{aligned}$$

where

$$f_j(t) = \frac{p_{\min}}{2}\left(\hat{r}_j(t) + \frac{1}{p_j(t)}\sqrt{\frac{\ln(N/\delta)}{KT}}\right).$$

Unraveling the recurrence, we rewrite $w_{\mathbf{a}}(t+1)$ by

$$\begin{aligned}w_{\mathbf{a}}(t+1) &= \exp\left(\sum_{\tau=1}^{t}\sum_b d_b(\tau) f_{a_b}(\tau)\right) \\ &= \exp\left(\sum_b \sum_{\tau=1}^{t} d_b(\tau) f_{a_b}(\tau)\right) \\ &= \prod_b g_{b,a_b}(t),\end{aligned}$$

implying that $w_{\mathbf{a}}(t+1)$ can be computed implicitly by maintaining the quantity $g_{b,j}(t) = \exp\left(\sum_{\tau=1}^{t} d_b(\tau) f_j(\tau)\right)$ for each b and j. Next, we compute W_t as follows: $W_t = \sum_{\mathbf{a}} w_{\mathbf{a}}(t) = \sum_{\mathbf{a}} \prod_b g_{b,a_b}(t) = \prod_b \left(\sum_j g_{b,j}(t)\right)$. Repeating the same trick, we have

$$\sum_{\mathbf{a}} \frac{w_{\mathbf{a}}(t)\boldsymbol{\xi}^{\mathbf{a}}(t)}{W_t} = \sum_b \frac{d_b(t) g_{b,j}(t)}{\sum_{j'=1}^{K} g_{b,j'}(t)},$$

which are the inputs to Algorithm 2 to produce the final arm-selection probabilities, $p_j(t)$ for all j. Therefore, for this structured set of experts, the time complexity of `Exp4.P` is only linear in K and B despite the exponentially large size of this set.

To compare algorithms, we collected historical user visit events with a random policy that chose articles uniformly at random for a fraction of user visits on the Yahoo! front page from May 1 to 9, 2009. This data contains over 41M user visits, a total of 253 articles, and about 21 candidate articles in the pool per user visit. (The pool of candidate articles changes over time, requiring corresponding modifications to `Exp4.P`[1]). With such random traffic data, we were able to obtain an unbiased *estimate of the CTR* (called **eCTR**) of a bandit algorithm as if it is run in the real world [15].

Due to practical concerns when applying a bandit algorithm, it is common to randomly assign each user visit to one of two "buckets": the *learning bucket*, where the bandit algorithm is run, and the *deployment bucket*, where the greedy policy (learned by the algorithm in the learning bucket) is used to serve users without receiving payoff information. Note that since the bandit algorithm continues to refine its policy based on payoff feedback in the learning bucket, its greedy policy may change over time. Its eCTR in the deployment bucket thus measures how good this greedy policy is. And as the deployment bucket is usually much larger than the learning bucket, the deployment eCTR is deemed a more important metric. Finally, to protect business-sensitive information, we only report *normalized eCTR*s, which are the actual eCTRs divided by the random policy's eCTR.

Based on estimates of T and K, we ran `Exp4.P` with $\delta = 0.01$. The same estimates were used to set γ in `Exp4` to minimize the regret bound in Theorem 7.1 of [2]. Table 1 summarizes eCTRs of all three algorithms in the two buckets. All differences are significant due to the large volume of data.

First, `Exp4.P`'s eCTR is slightly worse than `Exp4` in

[1] Our modification ensured that a new article's initial score was the average of all currently available ones'.

	Exp4.P	Exp4	ϵ-greedy
learning CTR	1.0525	1.0988	1.3827
deployment CTR	1.6512	1.5309	1.4290

Table 1: Overall click-through rates (eCTRs) of various algorithms on the May 1–9 data set.

the learning bucket. This gap is probably due to the more conservative nature of Exp4.P, as it uses the additional $\hat{v}_i$ terms to control variance, which in turn encourages further exploration. In return for the more extensive exploration, Exp4.P gained the highest deployment eCTR, implying its greedy policy is superior to Exp4.

Second, we note a similar comparison to the ϵ-greedy variant of Exp4.P. It was the most greedy among the three algorithms and thus had the highest eCTR in the learning bucket, but lowest eCTR in the deployment bucket. This fact also suggests the benefits of using the somewhat more complicated soft-max exploration scheme in Exp4.P.

Acknowledgments

We thank Wei Chu for assistance with the experiments and Kishore Papineni for helpful discussions.

This work was done while Lev Reyzin and Robert E. Schapire were at Yahoo! Research, NY. Lev Reyzin acknowledges this material is based upon work supported by the NSF under Grant #0937060 to the CRA for the Computing Innovation Fellowship program.

References

[1] Deepak Agarwal, Bee-Chung Chen, Pradheep Elango, Nitin Motgi, Seung-Taek Park, Raghu Ramakrishnan, Scott Roy, and Joe Zachariah. Online models for content optimization. In *Advances in Neural Information Processing Systems 21*, pages 17–24, 2008.

[2] Peter Auer, Nicolò Cesa-Bianchi, Yoav Freund, and Robert E. Schapire. The nonstochastic multiarmed bandit problem. *SIAM Journal of Computing*, 32(1):48–77, 2002.

[3] Peter Bartlett, Varsha Dani, Thomas Hayes, Sham Kakade, Alexander Rakhlin, and Ambuj Tewari. High-probability regret bounds for bandit online linear optimization. In *Conference on Learning Theory (COLT)*, 2008.

[4] Shai Ben-david, Dvid Pl, and Shai Shalev-shwartz. Agnostic online learning. In *Conference on Learning Theory*, 2009.

[5] Nicolo Cesa-Bianchi and Gabor Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, New York, NY, USA, 2006.

[6] David A. Freedman. On tail probabilities for martingales. *Annals of Probability*, 3(1):100–118, 1975.

[7] David Haussler and Philip M. Long. A generalization of Sauer's lemma. *J. Comb. Theory, Ser. A*, 71(2):219–240, 1995.

[8] David P. Helmbold and Robert E. Schapire. Predicting nearly as well as the best pruning of a decision tree. *Machine Learning*, 27(1):51–68, 1997.

[9] Leslie Pack Kaelbling. Associative reinforcement learning: Functions in k-DNF. *Machine Learning*, 15(3):279–298, 1994.

[10] Sham M. Kakade and Adam Kalai. From batch to transductive online learning. In *NIPS*, 2005.

[11] Sham M. Kakade, Shai Shalev-Shwartz, and Ambuj Tewari. Efficient bandit algorithms for online multiclass prediction. In *Proceedings of the Twenty-Fifth International Conference on Machine Learning (ICML)*, pages 440–447, 2008.

[12] Tze Leung Lai and Herbert Robbins. Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics*, 6:4–22, 1985.

[13] John Langford and Tong Zhang. The epoch-greedy algorithm for contextual multi-armed bandits. In *Neural Information Processing Systems (NIPS)*, 2007.

[14] A. Lazaric and R. Munos. Hybrid stochastic-adversarial on-line learning. In *Conference on Learning Theory*, 2009.

[15] Lihong Li, Wei Chu, John Langford, and Robert E. Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the Nineteenth International Conference on World Wide Web (WWW)*, 2010.

[16] Brendan McMahan and Matthew Streeter. Tighter bounds for multi-armed bandits with expert advice. In *Conference on Learning Theory (COLT)*, 2009.

[17] Herbert Robbins. Some aspects of the sequential design of experiments. *Bulletin of the American Mathematical Society*, 58(5):527–535, 1952.

[18] Alexander L. Strehl, Chris Mesterharm, Michael L. Littman, and Haym Hirsh. Experience-efficient learning in associative bandit problems. In *International Conference on Machine Learning (ICML)*, 2006.

[19] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction.* MIT Press, 1998.