Reduced-Variance Payoff Estimation in Adversarial Bandit Problems

Levente Kocsis and Csaba Szepesvári

Computer and Automation Research Institute of the
Hungarian Academy of Sciences, Kende u. 13-17, 1111 Budapest, Hungary
kocsis@sztaki.hu

Abstract. A natural way to compare learning methods in non-stationary environments is to compare their regret. In this paper we consider the regret of algorithms in adversarial multi-armed bandit problems. We propose several methods to improve the performance of the baseline exponentially weighted average forecaster by changing the payoff-estimation methods. We argue that improved performance can be achieved by constructing payoff estimation methods that produce estimates with low variance. Our arguments are backed up by both theoretical and empirical results. In fact, our empirical results show that significant performance gains are possible over the baseline algorithm.

1 Introduction

Regret is the excess cost incurred by a learner due to the lack of knowledge of the optimal solution. Since the notion of regret makes no assumptions on the environment, comparing algorithms by their regret represents an appealing choice for studying learning in non-stationary environments.

In this paper our focus is a slightly extended version of the adversarial bandit problem, originally proposed by Auer et. al [1]. The model that we start from is best described as a repeated game against an adversary using expert advice. In each round a player must choose an expert from a finite set of experts. In the given round the selected expert advices the player in playing a game against the adversary. At the end of the round the reward associated with the outcome of the game is communicated to the player. The player's goal is to maximize his total reward over the sequence of trials. Of course, the total reward depends on how strong the individual experts are and hence, a more reasonable criterion is to minimize the loss of the learner over the total reward of the best expert, i.e., the regret. If all the experts achieve a small total payoff then this goal is easy to achieve. However, if at least one the experts performs well then the algorithm must quickly identify this expert.

In this paper we are concerned with the performance of a particular class of algorithms built around the exponentially weighted average forecaster, Exp3 [3, 9, 1].[1] Despite the appealing theoretical guarantees that were derived beforehand

[1] Note that the although the basic setup allows for non-stationary environments, it is the algorithm designer's sole responsibility to come up with sufficiently strong experts. An alternative approach explored in [7] is to change the definition of regret

for this algorithm, little is known about its performance in real-world problems, our primary interest in this paper. In fact, our original interest was to apply on-line prediction and in particular Exp3 to opponent modelling in poker. Our rather unsatisfactory initial empirical results led us to the consideration of possible ways to improve the performance of Exp3.[2] The primary purpose of this paper is to show that such performance improvements are indeed possible. In particular, we propose several methods for this purpose.

In order to present the main idea underlying these constructions let us note that Exp3 works by constructing a payoff estimate for each of the experts and these estimates are used as the input of the exponentially weighted forecaster. The payoff estimates proposed in [1] have a specific form. Here, we argue for the importance of alternative payoff estimation methods that can exploit additional information often available to the player.

One such case that we consider here is when the experts are randomized and action probabilities are available to the player for *any* expert (not just the selected one). Another case is when additional side information (e.g. the cards) is available before each round. Under such assumptions we propose two alternative payoff estimation methods and compare the performance of the resulting algorithms with that of the baseline in a simpler domain (dynamic pricing) and in full poker. The results show that the alternative methods are capable of improving performance substantially. Our explanation of the improved performance is that the alternative payoff estimation methods give payoff estimates with lower (predictable) variance than the original estimate. In order to back up this hypothesis bounds are derived on the performance of a generalized form of Exp3 that explicitly include the (predictable) variance of the payoff estimates. The proofs are obtained by a careful modification of the original proof of [1] by replacing the (conservative) pointwise bounds of the second order quantities of the payoff estimates by their expectations at appropriate points. The importance of our results is that they show that it is possible to reduce the regret of the basic Exp3 algorithm by considering alternative payoff estimation methods.

The organization of the article is as follows: In Section 2 we introduce the framework, the notation and the basic algorithm, Exp3G, that is just Exp3 with generic payoff estimates. Our theoretical results are given in Section 3. The alternative payoff-estimation methods are presented in Section 4. Results in two domains, dynamic pricing and opponent modelling in Omaha Hi-Lo Poker are given in Section 5, whilst our conclusions are drawn in Section 6.

---

by allowing for the possibility that different experts (from a base set of experts) are used in different time-segments. Here, for the sake of simplicity we do not consider this case. However, we expect that our results generalize to this case without much difficulties.

[2] Such negative results have been documented recently, independently of us, in [6], but in a significantly simplified poker-variant.

2 Regret-minimization

We model on-line learning as a repeated game against an adversary with random payoffs. In our model the adversary is assumed to be oblivious, i.e. not allowed to adapt to the player, but otherwise is not restricted in any way.[3] In each time step the player may choose an expert from a finite set of experts. For simplicity, we label the experts by the integers $1, \ldots, N$. The protocol of the game is as follows: At time t, the environment is put in some state about which some information, C_t, is communicated to the player. The player then selects an expert, I_t, which in turn suggests an action A_t. Next, Nature generates some situation $Y_t \in \mathcal{Y}$. This situation Y_t may depend (randomly) on the sequence of past side information and situations, as well as time. Based on I_t and Y_t the player receives a payoff, $g_t = g(I_t, Y_t)$.

As an example of a game of this kind consider dynamic pricing with multiple products: Let $R(p_1, p_2, v)$ be the payoff of the vendor assuming that she selected the price p_1 and the customer selected the price p_2 and the value of the product to be sold is v. The particular form of R is not important for us, but for the sake of specificity choose e.g. $R(p_1, p_2) = (p_1 - v)\mathbb{I}(p_1 \leq p_2) - \alpha v \mathbb{I}(p_1 > p_2)$, where $\mathbb{I}(\text{true}) \equiv 1$ and $\mathbb{I}(\text{false}) \equiv 0$. Let us denote the price selected by expert i by $A_t^{(i)}$. Further, let B_t denote the price selected by the customer. Obviously, the payoff of the vendor in the tth step is $g_t = R(A_t^{I_t}, B_t, C_t)$. Hence, defining $Y_t = (C_t, B_t, A_t^{(1)}, \ldots, A_t^{(N)})$ and $g(i, c, b, a_1, \ldots, a_N) = R(a_i, b, c)$ we get that $g_t = g(I_t, Y_t)$ as expected.[4]

Denoting by $G_{i,n}$ the total payoff that the player would have received had she chosen the ith expert in *each* round and by $\hat{G}_n$ the actual payoff of the player, the goal of the player is to minimize the cumulative (external) regret

$$\max_i G_{i,n} - \hat{G}_n = \max_i \sum_{t=1}^{n} g(i, Y_t) - \sum_{t=1}^{n} g(I_t, Y_t). \tag{1}$$

2.1 The Exp3G Algorithm

We consider a generic version of the exponentially weighted average forecaster where our main assumption is that in each time-step, the player is

[3] The case when the adversary can adapt to the choices of the predictor was recently considered in [5], where it was noted that the performance of external regret-minimization algorithms can be arbitrarily far from the optimum. Extension of the present work to such problem is far from trivial (amongst other things since the definition of regret there is fundamentally different from the one considered here) and is left for future work. In our opinion, since many practical problems can be closely modelled as games against oblivious adversaries, the considered problem is still of sufficient interest.

[4] If the games played in the rounds are played in a reactive environment (like in poker), the ith expert's action $A_t^{(i)}$ will actually be a policy that governs the selection of the "low-level" actions. Likewise, B_t will be a policy of the environment, plus the additional necessary (often random) information (e.g. the sequence of random numbers used to draw actions for both the adversary and the player) that together fully determine the course of game.

capable of computing an unbiased estimate, $g'_t(i, Y_t)$, of the expected payoffs $\overline{g}_t(i, C_t) = \mathbb{E}[g(i, Y_t)|C^t, Y^{t-1}, I^{t-1}]$, $i = 1, \ldots, N$, where $Y^{t-1} = (Y_1, \ldots, Y_t)$, $I^{t-1} = (I_1, \ldots, I_{t-1})$, and C_t is the information received by the player.

Note that this assumption is weaker than assuming that the player is capable of computing an unbiased estimate of $g(i, Y_t)$, as we require only an estimate of $\overline{g}_t(i, C_t)$. Indeed, in many cases, such as in the above outlined dynamic pricing problem, it is not possible to obtain such an estimate.[5] In Section 4 we propose several methods to obtain estimators of $\overline{g}_t(i, C_t)$.

The generalized Exp3 algorithm (henceforth called Exp3G) is shown Figure 1. Exp3G is a straightforward generalization of the Exp3 algorithm of [1]: the main differences are that we allow for additional side-information and the payoff estimation procedure is left unspecified.[6] In particular, Exp3 is obtained if $g'_t(i, Y_t)$ is defined by

$$g'_t(i, Y_t) = \mathbb{I}(I_t = i)g(I_t, Y_t)/p_{I_t,t}, \tag{2}$$

where $p_{i,t}$ is the probability of choosing arm i in time step t. Further examples of various estimators will be given in subsequent sections: for the results of the next section details of these constructions are not needed.

Parameters: real numbers $0 < \eta, \gamma < 1$.
Initialization: $w_0 = (1, \ldots, 1)^T$;
For each round $t = 1, 2, \ldots$
(1) select an expert $I_t \in \{1, \ldots, N\}$ randomly according to

$$p_{i,t} = (1 - \gamma)\frac{w_{i,t-1}}{\sum_{k=1}^{N} w_{k,t-1}} + \frac{\gamma}{N};$$

(2) observe $g_t = g(I_t, Y_t)$;
(2) based on g_t, C_t, compute the feedbacks $g'_t(i, Y_t)$, $i = 1, \ldots, N$;
(3) compute $w_{i,t} = w_{i,t-1}e^{\eta g'_t(i,Y_t)}$.

Fig. 1. Exp3G: Generalized Exponentially Weighted Average Forecaster

3 Variance Dependent Regret Bounds

The key ingredient of our performance bound results is that the regret is bounded as a function of the predictable variance of the random feedbacks $g'_t(i, Y_t)$. Intuitively, it should be clear that the growth rate of regret should depend on this quantity, as shown in the following theorem that bounds the expected regret:

Theorem 1 *Consider algorithm Exp3G and assume that in each time step the random feedback $g'_t(i, Y_t)$ is an unbiased estimate of $g(i, Y_t)$, given C_t, I^{t-1}*

[5] In dynamic pricing this would require the knowledge of the price offered by the consumer, which, by assumption, is not available.

[6] Actually, the setup is also close to partial monitoring, where in each step a feedback vector is received and the main assumption is that based on this information the player can construct unbiased estimates of the payoffs of the experts [9].

and Y^{t-1}, and that the predictable variance of $g'_t(i, Y_t)$ can be bounded uniformly by σ^2: $\mathrm{Var}[g'_t(i, Y_t) \,|\, C_t, I^{t-1}, Y^{t-1}] \leq \sigma^2$. Further, let B be an upper bound on $|g'_t(i, Y_t)|$ and assume that $\mathbb{E}[g(i, Y_t)|C_t, I^{t-1}, Y^{t-1}] \leq 1$. Let $\overline{G}_{in} = \mathbb{E}[\sum_{t=1}^{n} g(i, Y_t)]$ be the expected cumulative gain assuming that option i is selected in each round, and let $\overline{\hat{G}}_n = \mathbb{E}[\sum_{t=1}^{n} g(I_t, Y_t)]$ denote the expected cumulative gain of Exp3G. Assume that $\eta \leq (\sqrt{5}-1)/(2B)$. Then

$$\max_i \overline{G}_{in} - \overline{\hat{G}}_n \leq \gamma n + \frac{\ln N}{\eta} + \eta n(1+\sigma^2). \tag{3}$$

Further, for $n \geq ((3-\sqrt{5})B^2 \ln N)/(2(1+\sigma^2))$, with the choice $\eta = \sqrt{\ln N/(n(1+\sigma^2))}$, and $\gamma = 0$, $\max_i \overline{G}_{in} - \hat{G}_n \leq \sqrt{(1+\sigma^2)n \ln N}$.

Note that under the conditions of the theorem the 'explicit' exploration term γ/N of $p_{i,t}$ can be eliminated without increasing the rate of the regret above $\sqrt{n}$. Actually, the upper bound is minimized when γ is zero.[7] Clearly, it is the assumption that the predictable variance of the estimates of the payoffs can be bounded *uniformly* that allows one to drop the exploration term. Indeed, in the case of partial monitoring studied by [9] and later by [4], this assumption does not necessarily hold with $\gamma = 0$ since then the option choice probabilities $p_{i,t}$ can become arbitrarily small and in these problems $g'_t(i, Y_t)$ is constructed by dividing the observed payoff by $p_{I_t,t}$.

Note that the bound scales with the bound on the predictable variance, σ as promised. However, the constant factor obtained with $\sigma = 0$ is $\sqrt{2}$ times larger than the best known bound for the full information case.

Proof. As it is usual in the study of exponentially weighted forecasters, we let $W_t = \sum_{i=1}^{N} w_{i,t}$ and consider the evolution of $\ln(W_t/W_{t-1})$. By letting $G'_{i,n} = \sum_{t=1}^{n} g'_t(i, Y_t)$ and using $\sum_{i=1}^{N} e^{\eta G'_{i,n}} \geq \max_i e^{\eta G'_{in}}$, due to the monotonicity of the logarithm function we get

$$\ln(W_n/W_0) \geq \eta \max_i G'_{in} - \ln N. \tag{4}$$

Now, let us bound $\ln(W_t/W_{t-1})$ from above. By our assumptions on η, $|\eta g'_t(i, Y_t)| \leq 1$. Exploiting the inequality $e^x \leq 1 + x + x^2$, which holds when $x \leq 1$ and $\ln(1+x) \leq x$, which holds when $x \geq -1$, elementary algebra yields

$$\ln \frac{W_t}{W_{t-1}} \leq \frac{\eta}{1-\gamma} \left(\sum_{i=1}^{N} p_{i,t} g'_t(i, Y_t) + \eta \sum_{i=1}^{N} p_{i,t} g'_t(i, Y_t)^2 \right).$$

Taking the sum of this expression w.r.t. t and combining the resulting inequality with (4) and reordering the terms gives $(1-\gamma)\max_i G'_{in} - \sum_{t,i} p_{i,t} g'_t(i, Y_t) \leq \frac{\ln N}{\eta} + \eta \sum_{t,i} p_{i,t} g'_t(i, Y_t)^2$. Now, using that the maximum of the expectation of some random variables is not larger than the expected value of

[7] Note that this does not mean that choosing $\gamma = 0$ gives the smallest regret. In fact, our observation is that $\gamma > 0$ often helps the algorithms.

their maxima, $\mathbb{E}[G_{i,n}] = \mathbb{E}[G'_{i,n}]$ and $\mathbb{E}[\sum_{t,i} p_{i,t} g'_t(i,Y_t)] = \overline{\hat{G}}_n$ (this follows by the assumptions that the random feedback $g'_t(i,Y_t)$ is an unbiased estimate of the expected value of $g(i,Y_t)$ given C_t, Y^{t-1} and I^{t-1}), one obtains $(1-\gamma)\max_{i=1,\dots,N} \overline{G}_{in} - \overline{\hat{G}}_n \le \frac{\ln N}{\eta} + \eta \mathbb{E}[\sum_{t,i} p_{i,t} g'_t(i,Y_t)^2]$. Hence by $\mathbb{E}[g'_t(i,Y_t)^2|C_t,H_{t-1}] = \mathrm{Var}[g'_t(i,Y_t)|C_t,H_{t-1}] + \mathbb{E}[g'_t(i,Y_t)|C_t,H_{t-1}]^2$, where $H_{t-1} = (Y^{t-1}, I^{t-1})$ denotes the history up to time t, the bound on the predictable variance on $g'_t(i,Y_t)$ and $|\mathbb{E}[g'_t(i,Y_t)|C_t,H_{t-1}]| = |\mathbb{E}[g(i,Y_t)|C_t,H_{t-1}]| \le 1$, we get that $\mathbb{E}[g'_t(i,Y_t)^2|C_t,H_{t-1}] \le \sigma^2 + 1$. Exploiting that by construction $p_{i,t}$ depends only on Y^{t-1}, I^{t-1} and does not depend on I_t, Y_t and $\sum_{i=1}^N p_{i,t} = 1$, we have that $\mathbb{E}[\sum_{t,i} p_{i,t} g'_t(i,Y_t)^2] \le n(1+\sigma^2)$. The bounds stated in the theorem now follow from $\overline{G}_{i,n} \le 1$.

Using the bounds of the previous theorem it is also possible to obtain bounds for the (random) regret defined in (1). Such bounds can be derived using versions of the Hoeffding and Bernstein maximal inequalities that work for bounded martingale difference series. In particular, the following result can be obtained:[8]

Theorem 2 *Assume that g'_t, g, satisfy the conditions stated in Theorem 1. Further, assume that $|g(i,Y_t)| \le 1$. Then, for any $\delta > 0$, $n \ge ((3-\sqrt{5})B^2 \ln N)/(2(1+\sigma^2))$ with the choice $\eta = \sqrt{\ln N/(n(1+\sigma^2))}$, the following bound on the regret of Exp3G holds with probability at least $1-\delta$:* $\max_i G_{in} - \hat{G}_n \le n^{1/2}\Big(((1+\sigma^2)\ln N)^{1/2} + (2+\sqrt{2}\sigma)\ln\big(\frac{N+1}{\delta}\big)^{1/2}\Big) + \frac{2(B+1)}{3}\ln\big(\frac{N+1}{\delta}\big)$.

By introducing an appropriate time dependent learning rate η_t and using the proof technique of [2] it is possible to derive a version of the above theorem that achieves the same order of regret uniformly in time. Then a simple application of the Borel-Cantelli lemma implies that under our conditions Exp3G is Hannan consistent, i.e., the average regret, $(1/n)(\max_{i=1,\dots,N} G_{i,n} - \hat{G}_n)$, converges to zero with probability one. Further, the rate of convergence is $O(n^{-1/2})$.[9]

4 Payoff Estimation Methods

In this section we give three construction for $g'_t(i,Y_t)$. Remember that the goal is to construct g'_t such that $\mathbb{E}[g'_t(i,Y_t)|C^t,H_{t-1}] = \mathbb{E}[g(i,Y_t)|C^t,H_{t-1}]$.

Likelihood Ratio Based Estimates For our first construction we assume that the experts are randomized and the action selection probabilities of any of the experts can be queried. The likelihood ratio based payoff estimation method

[8] The standard proof is omitted due to the lack of space.

[9] Note that in the case of the Exp3 algorithm the predictable variance of the payoff estimates will be roughly equal to $1/p_{i,t}$. Hence, in this case letting γ scale with $1/\sqrt{n}$ gives a variance that grows with the length of the period. A special construction that biases the estimates of the payoffs was introduced in [1] to control the variance of the payoff estimates. In our problems, where the variance is bounded by construction such a bias term is not needed. Actually, our experiments (not given here due to the lack of space) show that the regret increases with the bias term.

works as follows: Let the probability that action a is selected by expert i and given the side information c be denoted by $\pi_i(a|c)$[10] and consider

$$g'_t(i, Y_t) = \frac{\pi_i(A_t|C_t)}{\pi_{I_t}(A_t|C_t)} g(I_t, Y_t), \tag{5}$$

where A_t is the action selected by expert I_t in round t. Assume that the set of actions is finite and that $\pi_i(a|c) > 0$ for all i, a, c. Then, $\mathbb{E}[g'_t(i, Y_t)|C_t, H_{t-1}] = \sum_j p_{jt} \sum_a \mathbb{P}(A_t = a|C_t, I_t = j, H_{t-1}) \mathbb{E}[g'_t(i, Y_t)|C_t, I_t = j, A_t = a, H_{t-1}]$. Now, according to our assumptions $\mathbb{E}[g'_t(i, Y_t)|C_t, I_t = j, A_t = a, H_{t-1}]$ is well-defined (since $\pi_i(a|C_t) > 0$) and equals $\pi_i(a|C_t)/\pi_j(a|C_t) \mathbb{E}[g(j, Y_t)|C_t, I_t = j, A_t = a, H_{t-1}]$. Since $\mathbb{P}(A_t = a|C_t, I_t = j, H_{t-1})$ and $1/\pi_j(a, C_t)$ cancel each other, we get the desired equality. Let us further note that g'_t can be bounded by $\sup_{i,j,a,c} \pi_i(a|c)/\pi_j(a|c)$ and a uniform bound on the predictable variance of $g'_t(i, Y_t)$ can be derived provided that the predictable variance of $g(i, Y_t)$ is bounded (this follows when e.g. Y_t can take on finite values, or when $g(i, Y_t)$ is uniformly bounded as it was assumed in Theorem 2).

Let us make some remarks about the generality of this method. Assume for example that in each time step the payoff of the player results from following some policy in an episodic, multi-stage partially observable Markovian Decision Problem. Assume that the experts suggest some feedback policy. Then, it can be shown as e.g. in [8] that even when the player does not know the transition probabilities he is able to compute the appropriate likelihood ratios. Hence, this construction can be used e.g. in opponent modelling in (even unknown) Markov games. This will be exploited in our second experimental domain.

Reversed Importance Sampling: Algorithm LExp Motivated by the theoretical results of the previous section it looks a sensible idea to keep the predictable variance of $g'_t(i, Y_t)$ as small as possible. It is clear that the predictable variance can become large when the ratio $\pi_i(a|c)/\pi_{I_t}(a|c)$ is large. Now, let us observe that modifying the feedback by the said likelihood ratios can be thought of as a 'reversed' importance sampling: reversed in the sense that in this case it is not the sampling distribution that is controlled, but the function to be integrated. In importance sampling variance is reduced by drawing samples to those part of the domain where the function varies a lot (the optimal sampling density is proportional to $|f|$, where f is the function to be integrated). Since we cannot control the samples, we modify the function to be integrated so that it assumes large values where the samples concentrate and it becomes small (actually zero in the construction below) otherwise. This leads to the following modification of the likelihood weighting scheme:

Let $\phi_t(k, a, i) = \mathbb{I}(\pi_k(a|c)p_{k,t} < \pi_i(a|c)p_{i,t})$ and define

$$g'_t(i, Y_t) = \frac{(1 - \phi(I_t, A_t, i))}{\sum_{j=1}^N p_{j,t}(1 - \phi(j, A_t, i))} \frac{\pi_i(A_t|C_t)}{\pi_{I_t}(A_t|C_t)} g(I_t, Y_t). \tag{6}$$

The purpose of the modification is to make g'_t zero (small) for those 'rare' events when $\phi(I_t, A_t, i) = 1$. The "missing mass" must then be compensated

[10] The trivial extension when π depends on past information is omitted due to the lack of space.

for. This is achieved in the above construction by multiplying the feedbacks by $1/(\sum_{j=1}^{N} p_{j,t}(1-\phi(j,A_t,i)))$. Assuming sufficient regularity, one can show that this estimate satisfies the desired conditions. The algorithm that uses g'_t as defined above will be referred to in the description of the experiments as LExp. We note that the idea of nullifying/discounting feedbacks of rare events can be generalized to other estimation problems.

Compensation for the Expected Payoff: Algorithm CExp3 Another way to control the variance is to compensate the random feedbacks $g'_t(i,Y_t)$ for the expected payoff given the side information C_t. It should be clear that e.g. in dynamic pricing the product C_t controls to a large extent the distribution of the actual payoffs $g(i,Y_t)$. Hence, instead of the actual payoffs it makes sense to use the payoffs compensated for C_t. This can be achieved by defining $g^c(i,Y_t) = g(i,Y_t) - r(C_t)$, where $r(C_t)$ represents the mean payoff when seeing C_t. Similarly, $g'_t(i,Y_t)$ (of e.g. Equation 5) can be modified by subtracting $r(C_t)$ from it. This modification is meant to reduce the predictable variance $\mathrm{Var}[g'_t(i,Y_t)|Y^{t-1},I^{t-1}]$ of g'_t. An analysis entirely analogous to that of presented in Theorem 1 can be used to show that the bound on the actual regret does depend on this quantity, showing that compensating for the mean expected payoffs given side information is a reasonable strategy. Intuitively, the method works by compressing the range of payoffs. It should be clear that when a regret-minimization algorithm is run with the modified payoffs and if the algorithm is guaranteed to achieve a bound less than say K then the same bound applies to the original regret. This follows because $G^c_{i,n} \stackrel{\text{def}}{=} \sum_{t=1}^{n} g^c(i,Y_t) = G_{i,n} - \sum_{t=1}^{n} r(C_t)$ and $\hat{G}^c_n \stackrel{\text{def}}{=} \sum_{t=1}^{n} g^c(I_t,Y_t) = \hat{G}_n - \sum_{t=1}^{n} r(C_t)$ and thus $\max_i G^c_{i,n} - \hat{G}^c_n = \max_i G_{i,n} - \hat{G}_n$.[11] This algorithm will be referred to in the experiments as CExp3.

5 Experiments

The purpose of the experiments is to illustrate that the proposed method can indeed be used to improve the performance of Exp3. No claim is made on whether the algorithms considered for a particular domain represent the best fit: The domains simply serve to compare Exp3 with its descendants.[12] We will also show empirically that the estimates are unbiased and have reduced variance. In the experiments the parameters η, γ were tuned to minimize the regret of Exp3. The same set of parameters were then used for the competing alternatives of Exp3G.

[11] We note that in some cases it is possible to implement compensations without introducing any bias. This is the case for the multi-armed bandit problem where $g'_t(i,Y_t)$ can be replaced by e.g. $g'_t(i,Y_t) - (N-1)/N r(C_t)/p_{I_t,t}$ when $i = I_t$, and by $r(C_t)/N$ when $i \neq I_t$.

[12] In fact, both domains are stationary and stochastic. However, experiments with non-stationary versions of these problems yielded very similar results. In this paper we stick to the simpler domains to squeeze in the experiments into the limited space available. Results of the extended experiments will be given in the extended version of this paper available from the authors' homepages.

5.1 Experiments: Dynamic Pricing

In this section the performance of the proposed techniques is illustrated on the dynamic pricing problem with multiple products. In the particular instance that we consider here the vendor sets the price of the product, p_1, in the range of $[0, 1]$ and the customer decides to buy it or not. If the transaction occurred, the vendor receives a payoff equal to the price requested. Otherwise, the payoff is a fraction of the product value, $0.9v$, where the value of the product, v, is known to both parties.

In our experiments, the customer offers a price p_2, which is constructed by drawing a random number b from the Bernoulli distribution, $B(100, 0.5)$, and setting $p_2 = (b-50)/100+1.1v$. The vendor is advised by five experts that select prices at random according to some triangular densities. A parameter b controls the size of the support of the underlying densities (larger b means more randomness in the expert's suggestions). The first three experts use symmetric triangular densities with supports of size $2b$. The mean values of the underlying distributions are v, $1.1v$, and $v+2$ for expert one, two and three, respectively. The fourth and fifth experts use asymmetric triangular densities that are obtained from the symmetric ones by eliminating their left sides. The fourth expert chooses values in the range $[0.9v, 0.9v + b]$, whilst the last expert chooses values from the range $[v, v + b]$, with modes $0.9v + b$ and $v + b$, respectively.

We have experimented with three algorithms, Exp3, CExp3, and LExp. For CExp3 the payoff is compensated by subtracting v from the observed payoff. We considered two variants of the problem: in the first case the randomness of the experts' advice is low ($b = 0.05$), whilst in the second case randomness is high ($b = 0.3$). Table 1 gives the estimated expected value and standard deviation of $g'_t(i, Y_t)$ (in this case $C_t = v$, the expert index i corresponds to the columns) for the two problem variants. The expected values and variances of $g(i, Y_t)$ are also provided in the tables in the respective last rows. It can be readily observed that the estimated expected values are close to each other as expected (since the algorithms do not introduce any bias). In addition, both CExp3 and LExp reduce the variance of the estimates considerably as compared to Exp3. The average regret per game of the algorithms as a function of rounds is plotted in Figure 2. Notice that CExp3 beats Exp3 in all cases by a considerable margin. LExp performs similarly to Exp3 in the low-expert-noise ($b = 0.05$) case, whilst in the case of high expert-noise ($b = 0.3$) the performance of LExp approaches that of CExp3 and both beat Exp3 with a considerable margin. In particular, the difference in the performance of CExp3 and Exp3 is significant in both cases, whilst the difference between the performance of LExp and Exp3 is not significant in the first case and is significant in the second case at the level $p = 0.99$.

5.2 Experiments with Opponent Modelling in Omaha Hi-Lo Poker

In this section we study how the algorithms considered can be used for opponent modelling in a particular poker variant. Omaha Hi-Lo Poker is a card game played by two to ten players. At the start each player is dealt four private cards, and at later stages five community cards are dealt face up (three after the first,

$b = 0.05$					
$\mathbb{E}[g'_t(i,Y_t)]$	i=1	i=2	i=3	i=4	i=5
Exp3	0.388	0.399	0.401	0.371	0.396
CExp3	0.390	0.399	0.398	0.371	0.398
LExp	0.390	0.402	0.400	0.368	0.399
$\mathbb{E}[g(i,Y_t)]$	0.390	0.400	0.399	0.371	0.399

$b = 0.3$					
$\mathbb{E}[g'_t(i,Y_t)]$	i=1	i=2	i=3	i=4	i=5
Exp3	0.338	0.351	0.347	0.385	0.383
CExp3	0.343	0.354	0.348	0.381	0.382
LExp	0.343	0.356	0.351	0.383	0.384
$\mathbb{E}[g(i,Y_t)]$	0.343	0.356	0.350	0.383	0.384

$\sqrt{\mathrm{Var}[g'_t(i,Y_t)]}$	i=1	i=2	i=3	i=4	i=5
Exp3	1.782	1.435	1.427	2.097	1.573
CExp3	0.467	0.476	0.473	0.332	0.472
LExp	0.739	0.788	0.500	1.671	0.688
$\sqrt{\mathrm{Var}[g(i,Y_t)]}$	0.143	0.148	0.145	0.129	0.144

$\sqrt{\mathrm{Var}[g'_t(i;Y_t)]}$	i=1	i=2	i=3	i=4	i=5
Exp3	2.107	1.929	2.014	1.046	1.169
CExp3	0.735	0.724	0.726	0.744	0.745
LExp	0.856	0.573	0.651	0.475	0.412
$\sqrt{\mathrm{Var}[g(i,Y_t)]}$	0.153	0.151	0.150	0.141	0.143

Table 1. Monte-Carlo estimates of the payoffs, payoff estimates and their respective standard deviations for two instances of the dynamic pricing problem that use different sets of experts with small ($b = 0.05$) and large price variances ($b = 0.30$).

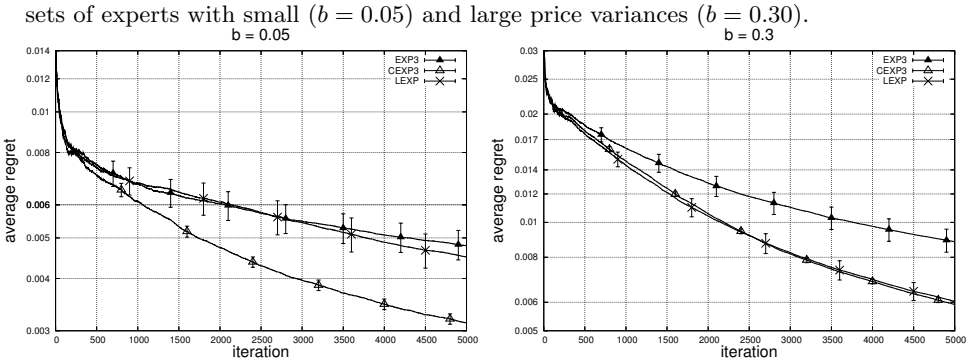

Fig. 2. Regret curves on two instances of the dynamic pricing problem. The 95% confidence intervals are also shown in the figures.

and one after the second and third betting round). In a betting round, the player on turn has three options: *fold*, *check/call*, or *bet/raise*. After the last betting round, the pot is split among the players depending on the strength of their cards. The pot is halved into a high side and a low side. For each side the players form a hand consisting of two private cards and three community cards. The high side is won according to the usual poker hand ranking. For the low side, a hand with five cards with different numerical values from Ace to eight has to be constructed. The winning low hand is the one with the lowest high card.

A natural performance measure of a player's strength is the average amount of money won per hand divided by the value of the small bet (sb/h). Typical differences between players are in the range of 0.05 to 0.2sb/h. Due to the randomness of cards, the payoff per game has a rather high variance that makes the evaluation of the performance, and thus any algorithm that operates based on the observed payoffs, rather slow. E.g. for showing that a 0.05sb/h difference is statistically significant in a two player game one has to play up to 20,000 games.

One method to significantly reduce the variance of the payoffs is to use *antithetic dealing* when in every second game each player is dealt the cards which his/her opponent had the game before, while the community cards are kept the same. As a result of this method the variance of payoffs is reduced by a factor

of ca. 6. Although it is not possible to use this method in real tournaments, we will use it in our experiments to obtain baseline results. Another way to reduce the variance is to compensate for the deal by subtracting from the payoff a value that depends on the strength of the hand in the context of the community cards and the hands of the opponents. Such a value can be estimated by playing the game with the same cards dealt to identical (robot) players (e.g. our poker playing program). The problem is neither can one use this technique in real tournaments where if a player folds then there is no way to replay the game (as the player's cards will be unknown). Still, in our simulated it is possible to use this method and hence it is also included for the sake of comparisons.

Opponent modelling is one of the most important aspects of poker. Our program, McRaise, uses its opponent model in assessing the probability that a particular betting sequence is played given a situation consisting of the private cards, community cards and the betting sequences of other opponents. In poker, it is rather common to classify (human) players according to their playing style. Similarly, we constructed six opponent models: *random* assumes a zero knowledge opponent (thus it will make no conclusions from the opponent's betting sequence), *greedy* assumes an opponent that plays according to the strength of his hand disregarding the play of its opponents, *smooth* is a smoother version of *greedy*, *mcr* is the generic opponent model used currently in McRaise, and takes into account most of the factors relevant to the game, *spsa* is an opponent model tuned against McRaise, and *humanoid* is based on the same information as *mcr* but expects cautious play, typical for most human players. For each of the opponent models our program assigns a probability to the possible actions given the current information in the game available to the player. These probabilities serve as the input to LExp.

In the experiments the performance of four algorithms, Exp3, aCExp3 (antithetic dealing), cCExp3 (card-strength compensated method), and LExp were investigated. As noted earlier, out of these four methods only Exp3 and LExp are suitable for real-world plays.

In the following we present the results for playing against McRaise. Tests were performed with two other opponents, with similar results obtained as those described below. The best expert in the case studied is *spsa* (+0.11sb/h), followed by *mcr* (0sb/h), *smooth* (-0.07sb/h), *greedy* (-0.12sb/h), *humanoid* (-0.22sb/h) and *random* (-0.77sb/h). The average regret in the course of learning for the four algorithms along with the probability of choosing *spsa* is plotted in Figure 3. Each algorithm detects *spsa* as the best expert at the end, but their convergence rates differ significantly. The two CExp3 variants converge much faster than Exp3, while LExp converges as fast as the better of the two (aCExp3). The average regret (of not playing with *spsa*) is hindered in the beginning by the exploration of the weaker experts (especially *random*), but for the three variance reduction methods the average per-round regret converges at a reasonably fast rate to zero. The differences in the performance of Exp3 and the improved methods are significant at the level $p = 0.99$. (The figures show error bars corresponding to 95% confidence intervals.)

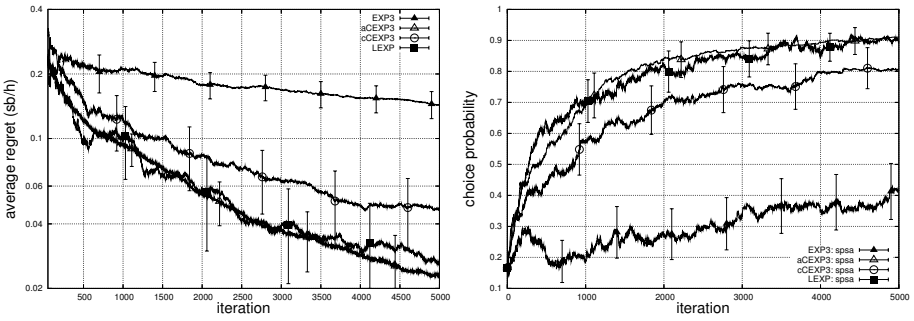

Fig. 3. Learning curves of the four algorithms. Left graph: regret, right graph: probability of choosing the best opponent model (*spsa*). Each data point is averaged over 100 runs.

6 Conclusions

In this paper we have considered regret-minimization via the use of a generalized form of the exponentially weighted average forecaster. We have argued that in certain problems alternative payoff estimation methods are possible that can reduce the variance of the payoff estimates, which, in turn may result in a decrease of the regret. Both our theoretical and empirical results show that the proposed methods are indeed effective in improving the performance of the baseline Exp3 algorithm.

References

1. P. Auer, N. Cesa-Bianchi, Y. Freund, and R.E. Schapire. The nonstochastic multi-armed bandit problem. *SIAM Journal on Computing*, 32:48–77, 2002.
2. P. Auer, N. Cesa-Bianchi, and C. Gentile. Adaptive and self-confident on-line learning algorithms. In *COLT-13*, pages 107–117. Morgan Kaufmann, San Francisco, 2000.
3. N. Cesa-Bianchi, Y. Freund, D. Haussler, D.P. Helmbold, R.E. Schapire, and M.K. Warmuth. How to use expert advice. *Journal of the ACM*, 44:427–485, 1997.
4. N. Cesa-Bianchi, G. Lugosi, and G. Stoltz. Regret minimization under partial monitoring. Preprint, 2004.
5. D. de Farias and N. Megiddo. Exploration-exploitation tradeoffs for experts algorithms in reactive environments. In *NIPS 17*, 2005.
6. B. Hoehn, F. Southey, R.C. Holte, and V. Bulitko. Effective short-term opponent exploitation in simplified poker. In *AAAI-2005*, 2005.
7. M.K. Warmuth M. Hebster. Tracking the best expert. *Machine Learning*, 32:151–178, 1998.
8. L. Peshkin and C.R. Shelton. Learning from scarce experience. In *ICML*, pages 498–505, 2002.
9. Antonio Piccolboni and Christian Schindelhauer. Discrete prediction games with arbitrary feedback and loss. In *COLT-15*, pages 208–223, 2001.