Exponential Weights on the Hypercube in Polynomial Time

Sudeep Raja Putta
University of Massachusetts Amherst

Abhishek Shetty
Microsoft Research India

Abstract

We study a general online linear optimization problem(OLO). At each round, a subset of objects from a fixed universe of n objects is chosen, and a linear cost associated with the chosen subset is incurred. To measure the performance of our algorithms, we use the notion of regret which is the difference between the total cost incurred over all iterations and the cost of the best fixed subset in hindsight. We consider Full Information and Bandit feedback for this problem. This problem is equivalent to OLO on the $\{0,1\}^n$ hypercube. The Exp2 algorithm and its bandit variant are commonly used strategies for this problem. It was previously unknown if it is possible to run Exp2 on the hypercube in polynomial time.

In this paper, we present a polynomial time algorithm called PolyExp for OLO on the hypercube. We show that our algorithm is equivalent Exp2 on $\{0,1\}^n$, Online Mirror Descent(OMD), Follow The Regularized Leader(FTRL) and Follow The Perturbed Leader(FTPL) algorithms. We show PolyExp achieves expected regret bound that is a factor of $\sqrt{n}$ better than Exp2 in the full information setting under L_∞ adversarial losses. Because of the equivalence of these algorithms, this implies an improvement on Exp2's regret bound in full information. We also show matching regret lower bounds. Finally, we show how to use PolyExp on the $\{-1,+1\}^n$ hypercube, solving an open problem in Bubeck et al (COLT 2012).

Proceedings of the 22[nd] International Conference on Artificial Intelligence and Statistics (AISTATS) 2019, Naha, Okinawa, Japan. PMLR: Volume 89.

1 Introduction

Consider the following abstract game which proceeds as a sequence of T rounds. In each round t, a player has to choose a subset S_t from a universe U of n objects. Without loss of generality, assume $U = \{1, 2, .., n\} = [n]$. Each object $i \in U$ has an associated loss $c_{t,i}$, which is unknown to the player and may be chosen by an adversary. On choosing S_t, the player incurs the cost $c_t(S_t) = \sum_{i \in S_t} c_{t,i}$. In addition the player receives some feedback about the costs of this round. The goal of the player is to choose the subsets such that the total cost incurred over a period of rounds is close to to the total cost of the best subset in hindsight. This difference in costs is called the *regret* of the player. Formally, regret is defined as:

$$\mathcal{R}_T = \sum_{t=1}^{T} c_t(S_t) - \min_{S \subseteq U} \sum_{t=1}^{T} c_t(S)$$

We can re-formulate the problem as follows. The 2^n subsets of U can be mapped to the vertices of the $\{0,1\}^n$ hypercube. The vertex corresponding to the set S is represented by its characteristic vector $X(S) = \sum_{i=1}^{n} 1\{i \in S\} e_i$. From now on, we will work with the hypercube instead of sets and use losses $l_{t,i}$ instead of costs. In each round, the player chooses $X_t \in \{0,1\}^n$. The loss vector l_t is be chosen by an adversary and is unknown to the player. The loss of choosing X_t is $X_t^\top l_t$. The player receives some feedback about the loss vector. The goal is to minimize regret, which is now defined as:

$$\mathcal{R}_T = \sum_{t=1}^{T} X_t^\top l_t - \min_{X \in \{0,1\}^n} \sum_{t=1}^{T} X^\top l_t$$

This is the *Online Linear Optimization(OLO)* problem on the hypercube. As the loss vector l_t can be set by an adversary, the player has to use some randomization in its decision process in order to avoid being foiled by the adversary. At each round $t = 1, 2, \ldots, T$, the player chooses an action X_t from the decision set $\{0,1\}^n$, using some internal randomization. Simultaneously, the adversary chooses a loss vector l_t, without access to the internal randomization of the player.

Since the player's strategy is randomized and the adversary could be adaptive, we consider the expected regret of the player as a measure of the player's performance. Here the expectation is with respect to the internal randomization of the player and the adversary's randomization.

We consider two kinds of feedback for the player.

1. *Full Information setting:* At the end of each round t, the player observes the loss vector l_t.

2. *Bandit setting:* At the end of each round t, the player only observes the scalar loss incurred $X_t^\top l_t$.

In order to make make quantifiable statements about the regret of the player, we need to restrict the loss vectors the adversary may choose. Here we assume that $||l_t||_\infty \leq 1$ for all t, also known as the L_∞ assumption.

There are three major strategies for online optimization, which can be tailored to the problem structure and type of feedback. Although, these can be shown to be equivalent to each other in some form, not all of them may be efficiently implementable. These strategies are:

1. Exponential Weights (EW)[10, 14]

2. Follow The Leader (FTL)[12]

3. Online Mirror Descent (OMD) [15].

For problems of this nature, a commonly used EW type algorithm is Exp2 [2, 3, 6]. For the specific problem of Online Linear Optimization on the hypercube, it was previously unknown if the Exp2 algorithm can be efficiently implemented [6]. So, previous works have resorted to using OMD algorithms for problems of this kind. The main reason for this is that Exp2 explicitly maintains a probability distribution on the decision set. In our case, the size of the decision set is 2^n. So a straightforward implementation of Exp2 would need exponential time and space.

1.1 Our Contributions

We use the following key observation: In the case of linear losses the probability distribution of Exp2 can be factorized as a product of n Bernoulli distributions. Using this fact, we design an efficient polynomial time algorithm called *PolyExp* for sampling sampling from and updating these distributions.

We show that PolyExp is equivalent to Exp2. In addition, we show that PolyExp is equivalent to OMD with entropic regularization and Bernoulli sampling. This allows us to analyze PolyExp's using powerful analysis techniques of OMD. We also show that PolyExp is equivalent to a Follow The Regularized Leader(FTRL) and a Follow The Perturbed Leader(FTPL) algorithms.

This kind of equivalence is rare. To the best of our knowledge, the only other scenario where this kind of equivalence holds is on the probability simplex for the so called experts problem.

In our paper, we focus on the L_∞ assumption. In the full information setting, directly analyzing Exp2 gives a regret bound of $O(n^{3/2}\sqrt{T})$. Using the equivalence to OMD, we show that PolyExp's regret bound is $n\sqrt{T}$. In the bandit setting however, PolyExp does not improve Exp2's analysis. These results are summarized by the table below.

L_∞		
	Full Information	Bandit
Exp2 [1]	$O(n^{3/2}\sqrt{T})$	$O(n^2\sqrt{T})$
PolyExp	$O(n\sqrt{T})$	$O(n^2\sqrt{T})$
Lowerbound	$\Omega(n\sqrt{T})$	$\Omega(n^{3/2}\sqrt{T})$

However, since we show that Exp2 and PolyExp are equivalent, they must have the same regret bound. This implies an improvement on Exp2's regret bound in full information.

Proposition 1. *For the Online Linear Optimization problem on the $\{0,1\}^n$ Hypercube, Exp2, OMD, FTRL, FTPL and PolyExp are equivalent. Moreover, under L_∞ adversarial losses, these algorithms have the following regret:*

1. *Full Information:* $O(n\sqrt{T})$

2. *Bandit:* $O(n^2\sqrt{T})$.

We also have the following lower bounds.

Proposition 2. *For the Online Linear Optimization problem on the $\{0,1\}^n$ Hypercube with L_∞ adversarial losses, the regret of any algorithm is at least:*

1. *Full Information:* $\Omega\left(n\sqrt{T}\right)$

2. *Bandit:* $\Omega(n^{3/2}\sqrt{T})$.

Finally, in [6], the authors state that it is not known if it is possible to sample from the exponential weights distribution in polynomial time for $\{-1,+1\}^n$ hypercube. We show how to use PolyExp on $\{0,1\}^n$ for $\{-1,+1\}^n$. We show that the regret of such an algorithm on $\{-1,+1\}^n$ will be a constant factor away

[1] direct analysis

from the regret of the algorithm on $\{0,1\}^n$. Thus, we can use PolyExp to obtain a polynomial time algorithm for $\{-1,+1\}^n$ hypercube.

We present the proofs of equivalence and regret of PolyExp within the main body of the paper. The remaining proofs are deferred to the appendix.

1.2 Relation to Previous Works

In previous works on OLO [9, 13, 2, 8, 6, 3] the authors consider arbitrary subsets of $\{0,1\}^n$ as their decision set. This is also called as Online Combinatorial optimization. In our work, the decision set is the entire $\{0,1\}^n$ hypercube. Moreover, the assumption on the adversarial losses are different. Most of the previous works use the L_2 assumption [6, 9, 8] and some use the L_∞ assumption [13, 2].

The Exp2 algorithm has been studied under various names, each with their own modifications and improvements. In its most basic form, it corresponds to the Hedge algorithm from [10] for full information. For combinatorial decision sets, it has been studied by [13] for full information. In the bandit case, several variants of Exp2 exist based on the exploration distribution used. These were studied in [9, 8] and [6]. It has been proven in [2] that Exp2 is provably sub optimal for some decision sets and losses.

Follow the Leader kind of algorithms were introduced by [12] for the full information setting, which can be extended to the bandit settings in some cases.

Mirror descent style of algorithms were introduced in [15]. For online learning, several works [1, 13, 6, 3] consider OMD style of algorithms. Other algorithms such as Hedge, FTRL and FTPL etc can be shown to be equivalent to OMD with the right regularization function and purturbation distribution. In fact, [18] show that OMD can always achieve a nearly optimal regret guarantee for a general class of online learning problems.

Under the L_∞ assumption, [13] and [2] present lower bounds that match our lower bounds. However, they prove that there exists a subset $S \subset \{0,1\}^n$ and a sequence of losses on S such that the regret is at least some lower bound. So, these results are not directly applicable in our case. So, we derive lower bounds specific for the entire hypercube, showing that there exists a sequence of losses on $\{0,1\}^n$ such that the regret is at least some lower bound.

We refer the readers to the books by [7], [5], [17], [11] and lectures by [16], [4] for a comprehensive survey of online learning algorithms.

2 Algorithms and Equivalences

In this section, we describe and analyze the Exp2, OMD with Entropic regularization and Bernoulli Sampling, and PolyExp algorithms and prove their equivalence.

2.1 Exp2

Algorithm: Exp2
Parameters: Learning Rate η
Let $w_1(X) = 1$ for all $X \in \{0,1\}^n$. For each round $t = 1, 2, \ldots, T$:

1. Sample X_t as below. Play X_t and incur the loss $X_t^\top l_t$.
   (a) Full Information: $X_t \sim p_t(X) = \frac{w_t(X)}{Z_t}$, where $Z_t = \sum_{Y \in \{0,1\}^n} w_t(Y)$
   (b) Bandit: $X_t \sim q_t(X) = (1-\gamma)p_t(X) + \gamma\mu(X)$. Here μ is the exploration distribution.
2. See Feedback and construct $\tilde{l}_t$.
   (a) Full Information: $\tilde{l}_t = l_t$.
   (b) Bandit: $\tilde{l}_t = P_t^{-1} X_t X_t^\top l_t$, where $P_t = \mathbb{E}_{X\sim q_t}[XX^\top]$
3. Update for all $X \in \{0,1\}^n$
$$w_{t+1}(X) = \exp\left(-\eta \sum_{\tau=1}^{t} X^\top \tilde{l}_\tau\right) \text{ or equivalently}$$
$$w_{t+1}(X) = \exp(-\eta X^\top \tilde{l}_t) w_t(X)$$

The loss vector used to update Exp2 must satisfy the condition that $\mathbb{E}_{X_t}[\tilde{l}_t] = l_t$. In the bandit case, the estimator was first proposed by [9]. Here, μ is the exploration distribution and γ is the mixing coefficient. We use uniform exploration over $\{0,1\}^n$.

Exp2 has several computational drawbacks. First, it uses 2^n parameters to maintain the distribution p_t. Sampling from this distribution in step 1 and updating it step 3 will require exponential time. For the bandit settings, even computing $\tilde{l}_t$ will require exponential time. We state the following regret bounds by analyzing Exp2 directly. The proofs are in the appendix. Later, we prove that these can be improved. These regret bounds are under the L_∞ assumption.

Theorem 3. *In the full information setting, if $\eta = \sqrt{\frac{\log 2}{nT}}$, Exp2 attains the regret bound:*
$$E[\mathcal{R}_T] \leq 2n^{3/2}\sqrt{T \log 2}$$

Theorem 4. *In the bandit setting, if* $\eta = \sqrt{\frac{\log 2}{9n^2T}}$ *and* $\gamma = 4n^2\eta$, *Exp2 with uniform exploration on* $\{0,1\}^n$ *attains the regret bound:*

$$\mathbb{E}[\mathcal{R}_T] \le 6n^2\sqrt{T\log 2}$$

2.2 PolyExp

Algorithm: PolyExp
Parameters: Learning Rate η
Let $x_{i,1} = 1/2$ for all $i \in [n]$. For each round $t = 1, 2, \dots, T$:

1. Sample X_t as below. Play X_t and incur the loss $X_t^\top l_t$.
   (a) Full information: $X_{i,t} \sim Bernoulli(x_{i,t})$
   (b) Bandit: With probability $1-\gamma$ sample $X_{i,t} \sim Bernoulli(x_{i,t})$ and with probability γ sample $X_t \sim \mu$
2. See Feedback and construct $\tilde{l}_t$
   (a) Full information: $\tilde{l}_t = l_t$
   (b) Bandit: $\tilde{l}_t = P_t^{-1}X_tX_t^\top l_t$, where $P_t = (1-\gamma)\Sigma_t + \gamma\mathbb{E}_{X\sim\mu}[XX^\top]$. The matrix Σ_t is $\Sigma_t[i,j] = x_{i,t}x_{j,t}$ if $i \neq j$ and $\Sigma_t[i,i] = x_i$ for all $i, j \in [n]$
3. Update for all $i \in [n]$:

$$x_{i,t+1} = \frac{1}{1+\exp(\eta\sum_{\tau=1}^{t}\tilde{l}_{i,\tau})} \text{ or equivalently}$$
$$x_{i,t+1} = \frac{x_{i,t}}{x_{i,t} + (1-x_{i,t})\exp(\eta\tilde{l}_{i,t})}$$

To get a polynomial time algorithm, we replace the sampling and update steps with polynomial time operations. PolyExp uses n parameters represented by the vector x_t. Each element of x_t corresponds to the mean of a Bernoulli distribution. It uses the product of these Bernoulli distributions to sample X_t and uses the update equation mentioned in step 3 to obtain x_{t+1}.

In the Bandit setting, we can sample X_t by sampling from $\prod_{i=1}^n Bernoulli(x_{t,i})$ with probability $1-\gamma$ and sampling from μ with probability γ. As we use the uniform distribution over $\{0,1\}^n$ for exploration, this is equivalent to sampling from $\prod_{i=1}^n Bernoulli(1/2)$. So we can sample from μ in polynomial time. The matrix $P_t = \mathbb{E}_{X\sim q_t}[XX^\top] = (1-\gamma)\Sigma_t + \gamma\Sigma_\mu$. Here Σ_t and Σ_μ are the covariance matrices when $X \sim \prod_{i=1}^n Bernoulli(x_{t,i})$ and $X \sim \prod_{i=1}^n Bernoulli(1/2)$ respectively. It can be verified that $\Sigma_t[i,j] = x_{i,t}x_{j,t}, \Sigma_\mu[i,j] = 1/4$ if $i \neq j$ and $\Sigma_t[i,i] = x_i, \Sigma_\mu[i,i] = 1/2$ for all $i, j \in [n]$. So P_t^{-1} can be computed in polynomial time.

2.3 Equivalence of Exp2 and PolyExp

We prove that running Exp2 is equivalent to running PolyExp.

Theorem 5. *Under linear losses* $\tilde{l}_t$, *Exp2 on* $\{0,1\}^n$ *is equivalent to PolyExp. At round* t, *The probability that PolyExp chooses* X *is* $\prod_{i=1}^n (x_{i,t})^{X_i}(1-x_{i,t})^{(1-X_i)}$ *where* $x_{i,t} = (1+\exp(\eta\sum_{\tau=1}^{t-1}\tilde{l}_{i,\tau}))^{-1}$. *This is equal to the probability of Exp2 choosing* X *at round* t, *ie:*

$$\prod_{i=1}^n (x_{i,t})^{X_i}(1-x_{i,t})^{(1-X_i)} = \frac{\exp(-\eta\sum_{\tau=1}^{t-1}X^\top\tilde{l}_\tau)}{Z_t}$$

where $Z_t = \sum_{Y\in\{0,1\}^n}\exp(-\eta\sum_{\tau=1}^{t-1}Y^\top\tilde{l}_\tau)$.

At every round, the probability distribution p_t in Exp2 is the same as the product of Bernoulli distributions in PolyExp. Lemma 15 is crucial in proving equivalence between the two algorithms. In a strict sense, Lemma 15 holds only because our decision set is the entire $\{0,1\}^n$ hypercube. The vector $\tilde{l}_t$ computed by Exp2 and PolyExp will be same. Hence, Exp2 and PolyExp are equivalent. Note that this equivalence is true for any sequence of losses as long as they are linear.

2.4 Online Mirror Descent

We present the OMD algorithm for linear losses on general finite decision sets. Our exposition is adapted from [5] and [17]. Let $\mathcal{X} \subset \mathbb{R}^n$ be an open convex set and $\bar{\mathcal{X}}$ be the closure of $\mathcal{X}$. Let $\mathcal{K} \in \mathbb{R}^d$ be a finite decision set such that $\bar{\mathcal{X}}$ is the convex hull of $\mathcal{K}$. The following definitions will be useful in presenting the algorithm.

Definition 6. ***Legendre Function:*** *A continuous function* $R : \bar{\mathcal{X}} \to \mathbb{R}$ *is Legendre if*

1. *R is strictly convex and has continuous partial derivatives on* $\mathcal{X}$.
2. $\lim_{x\to\bar{\mathcal{X}}/\mathcal{X}} \|\nabla R(x)\| = +\infty$

Definition 7. ***Legendre-Fenchel Conjugate:*** *Let* $R : \bar{\mathcal{X}} \to \mathbb{R}$ *be a Legendre function. The Legendre-Fenchel conjugate of* R *is:*

$$R^\star(\theta) = \sup_{x\in\mathcal{X}}(x^\top\theta - R(x))$$

Definition 8. ***Bregman Divergence:*** *Let* $R(x)$ *be a Legendre function, the Bregman divergence* $D_R : \bar{\mathcal{X}} \times \mathcal{X} \to \mathbb{R}$ *is:*

$$D_R(x\|y) = R(x) - R(y) - \nabla R(y)^\top(x-y)$$

Algorithm: Online Mirror Descent with Regularization $R(x)$
Parameters: Learning Rate η
Pick $x_1 = \arg\min_{x\in\bar{\mathcal{X}}} R(x)$. For each round $t = 1, 2, \ldots, T$:

1. Let p_t be a distribution on $\mathcal{K}$ such that $\mathbb{E}_{X\sim p_t}[X] = x_t$. Sample X_t as below and incur the loss $X_t^\top l_t$
   (a) Full information: $X_t \sim p_t$
   (b) Bandit: With probability $1-\gamma$ sample $X_t \sim p_t$ and with probability γ sample $X_t \sim \mu$.
2. See Feedback and construct $\tilde{l}_t$
   (a) Full information: $\tilde{l}_t = l_t$
   (b) Bandit: $\tilde{l}_t = P_t^{-1} X_t X_t^\top l_t$, where $P_t = (1-\gamma)\mathbb{E}_{X\sim p_t}[XX^\top] + \gamma\mathbb{E}_{X\sim\mu}[XX^\top]$.
3. Let y_{t+1} satisfy: $y_{t+1} = \nabla R^\star(\nabla R(x_t) - \eta\tilde{l}_t)$
4. Update $x_{t+1} = \arg\min_{x\in\bar{\mathcal{X}}} D_R(x||y_{t+1})$

2.5 Equivalence of PolyExp and Online Mirror Descent

For our problem, $\mathcal{K} = \{0,1\}^n$, $\bar{\mathcal{X}} = [0,1]^n$ and $\mathcal{X} = (0,1)^n$. We use entropic regularization:

$$R(x) = \sum_{i=1}^{n} x_i \log x_i + (1-x_i)\log(1-x_i)$$

This function is Legendre. The OMD algorithm does not specify the probability distribution p_t that should be used for sampling. The only condition that needs to be met is $\mathbb{E}_{X\sim p_t}[X] = x_t$, i.e, x_t should be expressed as a convex combination of $\{0,1\}^n$ and probability of picking X is its coefficient in the linear decomposition of x_t. An easy way to achieve this is by using Bernoulli sampling like in PolyExp. Hence, we have the following equivalence theorem:

Theorem 9. *Under linear losses $\tilde{l}_t$, OMD on $[0,1]^n$ with Entropic Regularization and Bernoulli Sampling is equivalent to PolyExp. The sampling procedure of PolyExp satisfies $\mathbb{E}[X_t] = x_t$. The update of OMD with Entropic Regularization is the same as PolyExp.*

In the bandit case, if we use Bernoulli sampling, $\mathbb{E}_{X\sim p_t}[XX^\top] = \Sigma_t$.

2.6 Regret of PolyExp via OMD analysis

Since OMD and PolyExp are equivalent, we can use the standard analysis tools of OMD to derive a regret bound for PolyExp. These regret bounds are under the L_∞ assumption.

Theorem 10. *In the full information setting, if $\eta = \sqrt{\frac{\log 2}{T}}$, PolyExp attains the regret bound:*

$$E[\mathcal{R}_T] \le 2n\sqrt{T\log 2}$$

We have shown that Exp2 on $\{0,1\}^n$ with linear losses is equivalent to PolyExp. We have also shown that PolyExp's regret bounds are tighter than the regret bounds that we were able to derive for Exp2 in full information. This naturally implies an improvement for Exp2's regret bounds as it is equivalent to PolyExp and must attain the same regret. However, in the bandit case PolyExp does not improve Exp2's regret bound. So, it has the same regret as Exp2 stated in Theorem 5.

2.7 Follow The Leader

PolyExp can be shown to be equivalent to a Follow The Regularized Leader(FTRL) and a Follow The Perturbed Leader(FTPL) algorithm. The FTRL algorithm can be easily deduced from the OMD algorithm as the Bregman projection step in OMD is not necessary in the case of entropic regularization. Hence, to derive the FTRL algorithm, we replace steps 3 and 4 in OMD with the following update step:

$$x_{t+1} = \arg\min_{x\in[0,1]^n}\left[\eta\sum_{\tau=1}^{t}\tilde{l}_\tau^\top x + R(x)\right]$$

In the FTPL algorithm, we draw a random vector v from some fixed distribution at each time step. We then choose the point which minimizes the sum of all previous losses and the perturbation vector v, by solving a simple linear optimization problem.

$$X_{t+1} = \arg\min_{x\in\{0,1\}^n}\left[\eta\sum_{\tau=1}^{t}\tilde{l}_\tau^\top x + v^\top x\right]$$

The FTPL algorithm is quite easy to implement in the full information setting as it folds the update and sampling steps into one computationally efficient step.

In the bandit setting however, we need to know the distribution $p_t(X)$, which is defined as follows:

$$p_t(X) = \Pr\left(X = \arg\min_{\{0,1\}^n}\left[\eta\sum_{\tau=1}^{t}\tilde{l}_\tau^\top x + v^\top x\right]\right)$$

This distribution may not always be computable.

We choose $F(x)$ to be the product of logistic distributions, ie distributions with CDF $\Pr(x \le \theta) = (1+\exp(-\theta))^{-1}$. In this case, we show that this FTPL is equivalent to PolyExp. Also, the distribution $p_t(X)$ has a simple closed form and $\mathbb{E}_{X\sim p_t}[XX^\top] = \Sigma_t$.

Algorithm: Follow the Perturbed Leader with Cumulative Distribution Function $F(x)$
Parameters: Learning Rate η
Pick $X_1 = \arg\min_{X \in \{0,1\}^n} v^\top x$, where v is drawn from distribution having CDF $F(x)$. For each round $t = 1, 2, \ldots, T$:

1. (a) Full information: Play X_t
   (b) Bandit: With probability $1-\gamma$ play X_t and with probability γ play $X_t \sim \mu$.
2. See Feedback and construct $\tilde{l}_t$
   (a) Full information: $\tilde{l}_t = l_t$
   (b) Bandit: $\tilde{l}_t = P_t^{-1} X_t X_t^\top l_t$, where $P_t = (1-\gamma)\mathbb{E}_{X\sim p_t}[XX^\top] + \gamma \mathbb{E}_{X\sim\mu}[XX^\top]$.
3. Update

$$X_{t+1} = \arg\min_{x\in\{0,1\}^n} \left[\eta \sum_{\tau=1}^{t} \tilde{l}_\tau^\top x + v^\top x\right]$$

Theorem 11. *Under linear losses $\tilde{l}_t$, FTRL with Entropic regularization and Bernoulli sampling, and FTPL with iid Logistic perturbations are equivalent to PolyExp.*

3 Comparison of Exp2's and PolyExp's regret proofs

Consider the results we have shown so far. We proved that PolyExp and Exp2 on the hypercube are equivalent. So logically, they should have the same regret bounds. But, our proofs say that PolyExp's regret is $O(\sqrt{n})$ better than Exp2's regret. What is the reason for this apparent discrepancy?

The answer lies in the choice of η and the application of the inequality $e^{-x} \le 1 + x - x^2$ in our proofs. This inequality is valid when $x \ge -1$. When analyzing Exp2, x is $\eta X^\top l_t = \eta L_t(X)$. So, to satisfy the constraints $x \ge -1$ we enforce that $|\eta L_t(X)| \le 1$. Since $|L_t(X)| \le n$, $\eta \le 1/n$. When analyzing PolyExp, x is $\eta l_{t,i}$ and we enforce that $|\eta l_{t,i}| \le 1$. Since we already assume $|l_{t,i}| \le 1$, we get that $\eta \le 1$. PolyExp's proof technique allows us to find a better η and achieve a better regret bound.

4 Lower bounds

We state the following lower bounds that establish the least amount of regret that any algorithm must incur. The lower bounds match the upper bounds of PolyExp proving that it is regret optimal. The proofs of the lower bounds can be found in the appendix.

Theorem 12. *For any learner there exists an adversary producing L_∞ losses such that the expected regret in the full information setting is:*

$$\mathbb{E}[\mathcal{R}_T] = \Omega\left(n\sqrt{T}\right).$$

Theorem 13. *For any learner there exists an adversary producing L_∞ losses such that the expected regret in the Bandit setting is:*

$$\mathbb{E}[\mathcal{R}_T] = \Omega\left(n^{3/2}\sqrt{T}\right).$$

5 $\{-1,+1\}^n$ Hypercube Case

Full information and bandit algorithms which work on $\{0,1\}^n$ can be modified to work on $\{-1,+1\}^n$. The general strategy is as follows:

1. Sample $X_t \in \{0,1\}^n$, play $Z_t = 2X_t - \mathbf{1}$ and incur loss $Z_t^\top l_t$.
   (a) Full information: $X_t \sim p_t$
   (b) Bandit: $X_t \sim q_t = (1-\gamma)p_t + \gamma\mu$
2. See feedback and construct $\tilde{l}_t$
   (a) Full information: $\tilde{l}_t = l_t$
   (b) Bandit: $\tilde{l}_t = P_t^{-1} Z_t Z_t^\top l_t$ where $P_t = \mathbb{E}_{X\sim q_t}[(2X-\mathbf{1})(2X-\mathbf{1})^\top]$
3. Update algorithm using $2\tilde{l}_t$

Theorem 14. *Exp2 on $\{-1,+1\}^n$ using the sequence of losses l_t is equivalent to PolyExp on $\{0,1\}^n$ using the sequence of losses $2\tilde{l}_t$. Moreover, the regret of Exp2 on $\{-1,1\}^n$ will equal the regret of PolyExp using the losses $2\tilde{l}_t$.*

Hence, using the above strategy, PolyExp can be run in polynomial time on $\{-1,1\}^n$ and since the losses are doubled its regret only changes by a constant factor.

6 Conclusions

For linear losses, we show that the Exp2 algorithm can be run on the hypercube in polynomial time using PolyExp. We also show equivalences to OMD, FTRL and FTPL. We improve Exp2's regret bound in full information using OMD's analysis, showing that it is minimax optimal. In the bandit setting, the regret bound could not be improved using this analysis. It remains to show how to achieve the minimax optimal regret in the bandit setting under L_∞ losses.

7 Proofs

7.1 Equivalence to Exp2

Lemma 15. *For any sequence of losses $\tilde{l}_t$, the following is true for all $t = 1, 2, .., T$:*

$$\prod_{i=1}^{n}(1+\exp(-\eta\sum_{\tau=1}^{t-1}\tilde{l}_{i,\tau})) = \sum_{Y\in\{0,1\}^n}\exp(-\eta\sum_{\tau=1}^{t-1}Y^\top\tilde{l}_\tau)$$

Proof. Consider $\prod_{i=1}^{n}(1+\exp(-\eta\sum_{\tau=1}^{t-1}\tilde{l}_{i,\tau}))$. It is a product of n terms, each consisting of 2 terms, 1 and $\exp(-\eta\sum_{\tau=1}^{t-1}\tilde{l}_{i,\tau})$. On expanding the product, we get a sum of 2^n terms. Each of these terms is a product of n terms, either a 1 or $\exp(-\eta\sum_{\tau=1}^{t-1}\tilde{l}_{i,\tau})$. If it is 1, then $Y_i = 0$ and if it is $\exp(-\eta\sum_{\tau=1}^{t-1}\tilde{l}_{i,\tau})$, then $Y_i = 1$. So,

$$\begin{aligned}\prod_{i=1}^{n}(1+\exp(-\eta\sum_{\tau=1}^{t-1}\tilde{l}_{i,\tau})) &= \sum_{Y\in\{0,1\}^n}\prod_{i=1}^{n}\exp(-\eta\sum_{\tau=1}^{t-1}\tilde{l}_{i,\tau})^{Y_i}\\ &= \sum_{Y\in\{0,1\}^n}\prod_{i=1}^{n}\exp(-\eta\sum_{\tau=1}^{t-1}\tilde{l}_{i,\tau}Y_i)\\ &= \sum_{Y\in\{0,1\}^n}\exp(-\eta\sum_{\tau=1}^{t-1}Y^\top\tilde{l}_\tau)\end{aligned}$$

□

Theorem 5. *Under linear losses $\tilde{l}_t$, Exp2 on $\{0,1\}^n$ is equivalent to PolyExp. At round t, The probability that PolyExp chooses X is $\prod_{i=1}^{n}(x_{i,t})^{X_i}(1-x_{i,t})^{(1-X_i)}$ where $x_{i,t} = (1+\exp(\eta\sum_{\tau=1}^{t-1}\tilde{l}_{i,\tau}))^{-1}$. This is equal to the probability of Exp2 choosing X at round t, ie:*

$$\prod_{i=1}^{n}(x_{i,t})^{X_i}(1-x_{i,t})^{(1-X_i)} = \frac{\exp(-\eta\sum_{\tau=1}^{t-1}X^\top\tilde{l}_\tau)}{Z_t}$$

where $Z_t = \sum_{Y\in\{0,1\}^n}\exp(-\eta\sum_{\tau=1}^{t-1}Y^\top\tilde{l}_\tau)$.

Proof. The proof is via straightforward substitution of the expression for $x_{i,t}$ and applying Lemma 15.

$$\begin{aligned}\prod_{i=1}^{n}(x_{i,t})^{X_i}(1-x_{i,t})^{(1-X_i)} &= \prod_{i=1}^{n}\frac{\left(\exp(\eta\sum_{\tau=1}^{t-1}\tilde{l}_{i,\tau})\right)^{1-X_i}}{1+\exp(\eta\sum_{\tau=1}^{t-1}\tilde{l}_{i,\tau})}\\ &= \prod_{i=1}^{n}\frac{\exp(-\eta\sum_{\tau=1}^{t-1}\tilde{l}_{i,\tau})^{X_i}}{1+\exp(-\eta\sum_{\tau=1}^{t-1}\tilde{l}_{i,\tau})}\\ &= \frac{\prod_{i=1}^{n}\exp(-\eta\sum_{\tau=1}^{t-1}\tilde{l}_{i,\tau}X_i)}{\prod_{i=1}^{n}(1+\exp(-\eta\sum_{\tau=1}^{t-1}\tilde{l}_{i,\tau}))}\\ &= \frac{\exp(-\eta\sum_{\tau=1}^{t-1}X^\top\tilde{l}_{i,\tau})}{\sum_{Y\in\{0,1\}^n}\exp(-\eta\sum_{\tau=1}^{t-1}Y^\top\tilde{l}_\tau)}\end{aligned}$$

□

7.2 Equivalence to OMD

Lemma 16. *The Fenchel Conjugate of $R(x) = \sum_{i=1}^{n}x_i\log x_i + (1-x_i)\log(1-x_i)$ is:*

$$R^\star(\theta) = \sum_{i=1}^{n}\log(1+\exp(\theta_i))$$

Proof. Differentiating $x^\top\theta - R(x)$ wrt x_i and equating to 0:

$$\begin{aligned}\theta_i - \log x_i + \log(1-x_i) &= 0\\ \frac{x_i}{1-x_i} &= \exp(\theta_i)\\ x_i &= \frac{1}{1+\exp(-\theta_i)}\end{aligned}$$

Substituting this back in $x^\top\theta - R(x)$, we get $R^\star(\theta) = \sum_{i=1}^{n}\log(1+\exp(\theta_i))$. It is also straightforward to see that $\nabla R^\star(\theta)_i = (1+\exp(-\theta_i))^{-1}$ □

Theorem 9. *Under linear losses $\tilde{l}_t$, OMD on $[0,1]^n$ with Entropic Regularization and Bernoulli Sampling is equivalent to PolyExp. The sampling procedure of PolyExp satisfies $\mathbb{E}[X_t] = x_t$. The update of OMD with Entropic Regularization is the same as PolyExp.*

Proof. It is easy to see that $E[X_{i,t}] = \Pr(X_{i,t} = 1) = x_{i,t}$. Hence $E[X_t] = x_t$.

The update equation is $y_{t+1} = \nabla R^\star(\nabla R(x_t) - \eta\tilde{l}_t)$. Evaluating ∇F and using $\nabla R^\star$ from Lemma 16:

$$\begin{aligned}y_{t+1,i} &= \frac{1}{1+\exp(-\log(x_{t,i})+\log(1-x_{t,i})+\eta\tilde{l}_{t,i})}\\ &= \frac{1}{1+\frac{1-x_{t,i}}{x_{t,i}}\exp(\eta\tilde{l}_{t,i})}\\ &= \frac{x_{t,i}}{x_{t,i}+(1-x_{t,i})\exp(\eta\tilde{l}_{t,i})}\end{aligned}$$

Since $0 \leq (1+\exp(-\theta))^{-1} \leq 1$, we have that $y_{i,t+1}$ is always in $[0,1]$. Bregman projection step is not required. So we have $x_{i,t+1} = y_{i,t+1}$ which gives the same update as PolyExp. □

7.3 Equivalence to FTRL and FTPL

Theorem 11. *Under linear losses $\tilde{l}_t$, FTRL with Entropic regularization and Bernoulli sampling, and FTPL with iid Logistic perturbations are equivalent to PolyExp.*

Proof. Using $R(x) = \sum_{i=1}^{n}x_i\log x_i + (1-x_i)\log(1-x_i)$ in FTRL, we can solve the optimization problem exactly:

$$\nabla_i \left(\eta \sum_{\tau=1}^{t} l_\tau^\top x + R(x) \right) = \eta \sum_{\tau=1}^{t} l_{i,\tau} + \log \frac{x_i}{1-x_i} = 0$$
$$\implies x_i = \frac{1}{1+\exp(\eta \sum_{\tau=1}^{t} l_{i,\tau})}$$

This is the same update equation as PolyExp.

For FTPL, the optimization problem outputs $X_i = 1$ if $\eta \sum_{\tau=1}^{t} l_{i,\tau} + v_i \leq 0$ and $X_i = 0$ otherwise. As v is a vector of iid random variables, we can conclude that $X_{i,t+1}$ are also independent. We have that:

$$\begin{aligned}\Pr(X_{i,t+1} = 1) &= \Pr(\eta \sum_{\tau=1}^{t} l_{i,\tau} + v_i \leq 0) \\ &= \Pr(v_i \leq -\eta \sum_{\tau=1}^{t} l_{i,\tau}) \\ &= \frac{1}{1+\exp(\eta \sum_{\tau=1}^{t} l_{i,\tau})}\end{aligned}$$

To obtain the last equality, we use the fact that v_i is drawn from the Logistic distribution. We can see this is equal to the probability of picking $X_{i,t+1} = 1$ in the PolyExp algorithm. □

7.4 PolyExp Full Information Regret Proof

Lemma 17 (see Theorem 5.5 in [5]). *For any $x \in \bar{\mathcal{X}}$, OMD with Legendre regularizer $R(x)$ with domain $\bar{\mathcal{X}}$ and $R^\star$ is differentiable on $\mathbb{R}^n$ satisfies:*

$$\sum_{t=1}^{T} x_t^\top l_t - \sum_{t=1}^{T} x^\top l_t \leq \frac{R(x) - R(x_1)}{\eta} + \frac{1}{\eta} \sum_{t=1}^{T} D_{R^\star}(\nabla R(x_t) - \eta l_t \| \nabla R(x_t))$$

Lemma 18. *If $|\eta l_{t,i}| \leq 1$ for all $t \in [T]$ and $i \in [n]$, OMD with entropic regularizer $R(x) = \sum_{i=1}^{n} x_i \log x_i + (1-x_i)\log(1-x_i)$ satisfies for any $x \in [0,1]^n$,:*

$$\sum_{t=1}^{T} x_t^\top l_t - \sum_{t=1}^{T} x^\top l_t \leq \frac{n \log 2}{\eta} + \eta \sum_{t=1}^{T} x_t^T l_t^2$$

Proof. We start from Lemma 17. Using the fact that $x\log(x) + (1-x)\log(1-x) \geq -\log 2$, we get $R(x) - R(x_1) \leq n \log 2$. Next we bound the Bregmen term using Lemma 16

$$D_{R^\star}(\nabla R(x_t) - \eta l_t \| \nabla R(x_t)) = R^\star(\nabla R(x_t) - \eta l_t) - R^\star(\nabla R(x_t)) + \eta l_t^\top \nabla R^\star(\nabla R(x_t))$$

Using that fact that $\nabla R^\star = (\nabla R)^{-1}$, the last term is $\eta x_t^\top l_t$. The first two terms can be simplified as:

$$\begin{aligned}&R^\star(\nabla R(x_t) - \eta l_t) - R^\star(\nabla R(x_t)) \\ &= \sum_{i=1}^{n} \log \frac{1+\exp(\nabla R(x_t)_i - \eta l_{t,i})}{1+\exp(\nabla R(x_t)_i)} \\ &= \sum_{i=1}^{n} \log \frac{1+\exp(-\nabla R(x_t)_i + \eta l_{t,i})}{\exp(\eta l_{t,i})(1+\exp(-\nabla R(x_t)_i)}\end{aligned}$$

Using the fact that $\nabla R(x_t)_i = \log x_i - \log(1-x_i)$:

$$\begin{aligned}&= \sum_{i=1}^{n} \log \frac{x_{t,i} + (1-x_{t,i})\exp(\eta l_{t,i})}{\exp(\eta l_{t,i})} \\ &= \sum_{i=1}^{n} \log(1 - x_{t,i} + x_{t,i}\exp(-\eta l_{t,i}))\end{aligned}$$

Using the inequality: $e^{-x} \leq 1 - x + x^2$ when $x \geq -1$. So when $|\eta l_{t,i}| \leq 1$:

$$\leq \sum_{i=1}^{n} \log(1 - \eta x_{t,i} l_{t,i} + \eta^2 x_{t,i} l_{t,i}^2)$$

Using the inequality: $\log(1-x) \leq -x$

$$\leq -\eta x_t^\top l_t + \eta^2 x_t^\top l_t^2$$

The Bregman term can be bounded by $-\eta x_t^\top l_t + \eta^2 x_t^\top l_t^2 + \eta x_t^\top l_t = \eta^2 x_t^\top l_t^2$ Hence, we have:

$$\sum_{t=1}^{T} x_t^\top l_t - \sum_{t=1}^{T} x^\top l_t \leq \frac{n \log 2}{\eta} + \eta \sum_{t=1}^{T} x_t^T l_t^2$$

□

Theorem 10. *In the full information setting, if $\eta = \sqrt{\frac{\log 2}{T}}$, PolyExp attains the regret bound:*

$$E[\mathcal{R}_T] \leq 2n\sqrt{T \log 2}$$

Proof. Applying expectation with respect to the randomness of the player to definition of regret, we get:

$$\begin{aligned}\mathbb{E}[\mathcal{R}_T] &= \mathbb{E}[\sum_{t=1}^{T} X_t^\top l_t - \min_{X^\star \in \{0,1\}^n} \sum_{t=1}^{T} X^{\star\top} l_t] \\ &= \sum_{t=1}^{T} x_t^\top l_t - \min_{X^\star \in \{0,1\}^n} \sum_{t=1}^{T} X^{\star\top} l_t\end{aligned}$$

Applying Lemma 18, we get $E[\mathcal{R}_T] \leq \frac{n \log 2}{\eta} + \eta \sum_{t=1}^{T} x_t^T l_t^2$. Using the fact that $|l_{i,t}| \leq 1$, we get $\sum_{t=1}^{T} x_t^T l_t^2 \leq nT$.

$$\mathbb{E}[\mathcal{R}_T] \leq \eta n T + \frac{n \log 2}{\eta}$$

Optimizing over the choice of η, we get that the regret is bounded by $2n\sqrt{T \log 2}$ if we choose $\eta = \sqrt{\frac{\log 2}{T}}$. □

References

[1] J. D. Abernethy, E. Hazan, and A. Rakhlin. Competing in the dark: An efficient algorithm for bandit linear optimization. 2009.

[2] J.-Y. Audibert, S. Bubeck, and G. Lugosi. Minimax policies for combinatorial prediction games. In *Proceedings of the 24th Annual Conference on Learning Theory*, pages 107–132, 2011.

[3] J.-Y. Audibert, S. Bubeck, and G. Lugosi. Regret in online combinatorial optimization. *Mathematics of Operations Research*, 39(1):31–45, 2013.

[4] S. Bubeck. Introduction to online optimization. *Lecture Notes*, pages 1–86, 2011.

[5] S. Bubeck and N. Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends® in Machine Learning*, 5(1):1–122, 2012.

[6] S. Bubeck, N. Cesa-Bianchi, and S. Kakade. Towards minimax policies for online linear optimization with bandit feedback. In *Annual Conference on Learning Theory*, volume 23, pages 41–1. Microtome, 2012.

[7] N. Cesa-Bianchi and G. Lugosi. *Prediction, learning, and games*. Cambridge university press, 2006.

[8] N. Cesa-Bianchi and G. Lugosi. Combinatorial bandits. *Journal of Computer and System Sciences*, 78(5):1404–1422, 2012.

[9] V. Dani, S. M. Kakade, and T. P. Hayes. The price of bandit information for online optimization. In *Advances in Neural Information Processing Systems*, pages 345–352, 2008.

[10] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139, 1997.

[11] E. Hazan. Introduction to online convex optimization. *Foundations and Trends® in Optimization*, 2(3-4):157–325, 2016.

[12] A. Kalai and S. Vempala. Efficient algorithms for online decision problems. *Journal of Computer and System Sciences*, 71(3):291–307, 2005.

[13] W. M. Koolen, M. K. Warmuth, and J. Kivinen. Hedging structured concepts. In *COLT*, pages 93–105. Citeseer, 2010.

[14] N. Littlestone and M. K. Warmuth. The weighted majority algorithm. *Information and computation*, 108(2):212–261, 1994.

[15] A. S. Nemirovsky and D. B. Yudin. Problem complexity and method efficiency in optimization. 1983.

[16] A. Rakhlin and A. Tewari. Lecture notes on online learning. *Draft, April*, 2009.

[17] S. Shalev-Shwartz. Online learning and online convex optimization. *Foundations and Trends® in Machine Learning*, 4(2):107–194, 2012.

[18] N. Srebro, K. Sridharan, and A. Tewari. On the universality of online mirror descent. In *Advances in neural information processing systems*, pages 2645–2653, 2011.