Budgeted Prediction With Expert Advice

Kareem Amin*
University of Pennsylvania,
Philadelphia, PA
akareem@seas.upenn.edu

Satyen Kale*
Yahoo Labs
New York, NY
satyen@yahoo-inc.com

Gerald Tesauro
IBM T.J. Watson Research Center
Yorktown Heights, NY
gtesauro@us.ibm.com

Deepak Turaga
IBM T.J. Watson Research Center
Yorktown Heights, NY
turaga@us.ibm.com

February 9, 2015

Abstract

We consider a budgeted variant of the problem of learning from expert advice with N experts. Each queried expert incurs a cost and there is a given budget B on the total cost of experts that can be queried in any prediction round. We provide an online learning algorithm for this setting with regret after T prediction rounds bounded by $O\left(\sqrt{\frac{C}{B}\log(N)T}\right)$, where C is the total cost of all experts. We complement this upper bound with a nearly matching lower bound $\Omega\left(\sqrt{\frac{C}{B}T}\right)$ on the regret of any algorithm for this problem. We also provide experimental validation of our algorithm.

1 Introduction

In many real world domains, systems must make real-time predictions, but have natural contraints on the resources that can be devoted to making such a prediction. For example, an autonomous vehicle may need to predict the future location of objects in its environment in order to avoid collision. Since slight changes in lighting or reflection can drastically change the optimal choice of model for object-tracking (see [9, 15]), this model selection must happen while the system is online. At the same time the real-time characteristics of such systems imply a natural constraint (i.e., a "budget") on machines resources (processing cycles, memory, bandwidth, etc.) that can be devoted to making a prediction. In other real world applications such as health care and financial trading, there may also be explicit financial costs that limit acquisition of data relevant to making predictions.

One might hope to make predictions in the above scenarios using an online learning algorithm, such as learning from expert advice. However, in real systems "experts" are frequently just learned models themselves. Budget constraints can therefore preclude the possibility of using a traditional

*Work done while the authors were at IBM T.J. Watson Research.

experts algorithm with a large number of experts. In the full information setting, running an experts algorithm might involve evaluating each and every model, draining machine resources.

There are two trivial, but suboptimal solutions. One solution is to simply cull the total number of models. This decreases the performance that one might hope to ever achieve: by removing experts, the performance of the best expert has worsened. The other trivial solution is to use a large number of models N, but to only request a prediction from a single model on every round. Since only a single prediction is observed, only the loss of a single expert/model can be computed, and we are in the so-called bandit information setting. While computationally efficient, this comes at a cost of $\sqrt{N}$ cumulative regret. Moreoever, such an approach may be *too* conservative with respect to resource consumption: if the system is able to accomodate more consumption, a bandit approach leaves unused budget on the table.

Motivated by these settings, we study a bounded cost variant of the problem of learning from expert advice problem [7]. In this variant, there is a specified cost for obtaining the advice of a given expert. There is a budget on the total cost of experts that can be queried in any prediction round. An online learning algorithm queries a subset of experts (respecting the budget constraint) for advice on each round, uses this advice to make its own prediction, and suffers the associated loss. The algorithm also observes losses of the queried experts (and gets no information on the loss of the unqueried experts). The goal of the algorithm is to minimize regret, where regret is defined, as is standard in online learning, to be the difference between the total loss incurred by the algorithm and the loss of the best expert in hindsight.

In the first part of this paper, we prove our main theoretical results, stated below. We then demonstrate our algorithm's performance on real data in a setting emulating the aforementioned motivation for this work.

Theorem 1 *In the setting where we have access to N experts with total cost C and budget $B \leq C$, there is an online learning algorithm whose expected regret is bounded by $O\left(\sqrt{\frac{C}{B}\log(N)T}\right)$ after T prediction rounds. Furthermore, any algorithm for the problem must incur expected regret of $\Omega\left(\sqrt{\frac{C}{B}T}\right)$ on some sequence of expert advices and losses.*

The regret bound proved here interpolates between two extreme cases. On the one hand, if $B = C$, then we get the standard experts setting, and the regret bound reduces to the well-known $O(\sqrt{\log(N)T})$ bound of Multiplicative Weights (MW)/Hedge algorithm [7]. On the other hand, if all the costs are the same, and the budget is the cost of any single expert, then we get the standard multiarmed bandits setting, and the regret bound reduces to the well-known $O(\sqrt{N\log(N)T})$ bound of the EXP3 algorithm of [3].

2 Related work

Several authors have considered sequential learning problems in which there are budget constraints different from ours. In the setting of [4, 13], the learner is restricted to the bandit setting, but consumes a resource upon selecting an action, for which the learner has a fixed budged over all rounds. The goal is to balance exploration, exploitation and the future use of the resource. [16] considers a setting in which observations are costly, but are incorporated in the algorithm's loss. Furthermore, learning from limited observations of attributes is an important theme in offline methods [8, 5, 10].

A special case of the problem considered in this paper has appeared in the work of [12]. This special case is the setting where all experts have the same cost of 1. The regret they obtain for this special case matches our regret bound specialized to the unit cost case. Their algorithm is quite different from ours, however, and uniformly selects experts to query, while we use a binning technique which naturally works for non-uniform costs. Our approach has the following benefits over the algorithm in [12]: (a) it is unclear how to generalize their algorithm to non-uniform costs[1] (b) our approach easily generalizes to the case when both the costs and the budget can change with time (see Section 6) and (c) our approach readily yields a weighted averaging scheme for the case of convex losses (see Section 6) which yields significant performance benefits in practice.

A similar binning strategy was used in the work of [11] which considers the budgeted prediction problem in the multiarmed bandit setting, with uniform costs. The techniques in this paper allow generalization of the results of [11] to the non-uniform case, but we omit the details for brevity.

Another related work is [1] which considers a setting where the learner has partial observability of expert losses specified by an observability graph on the experts. For any expert, using their advice also yields information about the advice of their neighboring experts in the observability graph. The binning idea used in this paper can be used to construct an observability graph consisting of disjoint cliques of experts that are in the same bin. Applying the algorithm of [1] then yields similar regret bounds to the ones appearing in this paper. However, we emphasize that the budgeted prediction problem doesn't come endowed with an observability graph structure, and using the binning technique is crucial in obtaining the regret bounds in this paper. Furthermore the lower bound on regret in [1] doesn't apply to the budgeted prediction problem again because of the lack of an observability graph structure.

3 Preliminaries

We are given a set $\mathcal{H}$ of N experts. In each round t, for $t = 1, 2, \ldots, T$, an adversary sets losses $\ell_t(h) \in [0, 1]$ for the predictions of each expert $h \in \mathcal{H}$. Expert h's prediction can be requested at cost c_h per round. Let $C := \sum_{h \in \mathcal{H}} c_h$. There is a per-round budget B on the total cost of experts queried in each round. In each round t, the learner queries a subset of experts S_t of total cost at most B, chooses one of the experts h_t in S_t, makes that expert's prediction, and suffers the same loss as the chosen expert, viz. $\ell_t(h_t)$. The learner observes the losses of all the experts in S_t. The goal is to minimize the expected regret with respect to the loss of the best expert, where the regret is defined as:

$$\text{Regret}_T := \sum_{t=1}^{T} \ell_t(h_t) - \min_{h \in \mathcal{H}} \sum_{t=1}^{T} \ell_t(h).$$

In case the learner makes its predictions randomly, we consider the expected regret instead.

4 Algorithm

The algorithm works as follows. In the beginning, partition the experts in to as few "bins" of total cost at most B as possible: it is well-known that $\frac{2C}{B}$ bins suffice [14], say using the simple

[1]Simply generalizing their algorithm by creating as many copies of each expert as their cost and then treating each copy as having unit cost fails because the while sampling unit cost copies of experts may respect the budget, actually using those experts may end up having higher cost than the budget.

greedy first-fit algorithm. Run the Multiplicative Weights (MW)/Hedge algorithm [7, 2] for learning from expert advice. This algorithm keeps a weight for every expert which induces a probability distribution on the experts via normalization. In each round, the algorithm picks an expert from the induced probability distribution, and queries all experts in the bin of the selected expert. This ensures that the total cost of experts queried in any round is at most B. The algorithm then follows the advice of the chosen expert, suffers the same loss as the chosen expert, and observes the losses of all experts in the chosen bin. It then updates the weights of all experts using a standard multiplicative weights update rule with an unbiased estimator for the loss of each expert in the exponent of the multiplicative factor. These loss estimators are non-zero only for experts in the chosen bin; thus they can be computed for all experts and the algorithm is well-defined. The pseudo-code[2] is given in Algorithm 1.

Algorithm 1 Budgeted Experts Algorithm (BEXP).

Require: Learning rate parameter η

Partition the experts in to as few "bins" as possible, each with total cost at most B. Let the number of bins used be R. Call the bins $B_1, B_2, \dots, B_R$, and define $\mathcal{R} := \{1, 2, \dots, R\}$.

Initialize distribution q_1 to be the uniform distribution over the N experts.

for $t = 1, 2, \dots, T$ **do**

Sample an expert $h_t \sim q_t$, let $I_t \in \mathcal{R}$ be the index of the bin to which h_t belongs, and set S_t to B_{I_t}.

Query the advice of all experts in S_t.

Make the same prediction as h_t and incur loss $\ell_t(h_t)$, and observe the loss of experts $h \in S_t$.

Compute loss estimators for all experts h as follows:

$$\hat{\ell}_t(h) \;:=\; \begin{cases} \frac{\ell_t(h)}{r_t(I_t)} & \text{if } h \in S_t \\ 0 & \text{otherwise,} \end{cases} \tag{1}$$

where r_t is the probability distribution over bin indices $\mathcal{R}$ induced by q_t, i.e. for $i \in \mathcal{R}$, $r_t(i) = \sum_{h \in B_i} q_t(h)$.

Update the distribution:

$$q_{t+1}(h) \;:=\; q_t(h) \exp(-\eta \hat{\ell}_t)(h)/Z_t,$$

where Z_t is the constant required to make q_{t+1} a distribution, i.e. $\sum_h q_{t+1}(h) = 1$.

end for

5 Analysis

We first prove a couple of utility lemmas. The first lemma shows that the loss estimators we construct are unbiased for all experts:

Lemma 1 *For all rounds t and all experts h, we have $\mathbb{E}[\hat{\ell}_t(h)] = \ell_t(h)$.*

[2] We use a fixed learning rate here for simplicity; the extension to time dependent learning rates which do not need knowledge of T is standard.

Proof: For any expert h, let B_i be the bin it belongs to. The probability that B_i is picked in any round t is exactly $r_t(i)$. Thus, $\mathbb{E}_t[\hat{\ell}_t(h)] = \frac{\ell_t(h)}{r_t(i)} \cdot r_t(i) + 0 \cdot (1 - r_t(i)) = \ell_t(h)$. Taking expectation over all the randomness up to time $t-1$, the proof is complete. □

The next lemma gives a bound on the variance of the estimated losses:

Lemma 2 *For all rounds t we have $\mathbb{E}[\sum_h q_t(h)\hat{\ell}_t(h)^2] \leq R$.*

Proof: Conditioning on the value of I_t, we can upper bound $\sum_h q_t(h)\hat{\ell}_t(h)^2$ as follows:

$$\sum_h q_t(h)\hat{\ell}_t(h)^2 \;=\; \sum_{h \in B_{I_t}} q_t(h)\left(\frac{\ell_t(h)}{r_t(I_t)}\right)^2 \;\leq\; \frac{1}{r_t(I_t)},$$

since $\sum_{h \in B_{I_t}} q_t(h) = r_t(I_t)$ and $\ell_t(h)^2 \leq 1$. Therefore,

$$\mathbb{E}_t\left[\sum_h q_t(h)\hat{\ell}_t(h)^2\right] \;\leq\; \mathbb{E}_t\left[\frac{1}{r_t(I_t)}\right] \;=\; \sum_{i=1}^{R} r_t(i) \cdot \frac{1}{r_t(i)} \;=\; R.$$

Taking expectation over all the randomness up to time $t-1$, the proof is complete. □

Finally, we give a regret bound for the algorithm:

Theorem 2 *Set $\eta = \sqrt{\frac{2\log(N)}{RT}}$. Then the expected regret of the algorithm is bounded by $\sqrt{2R\log(N)T} \leq 2\sqrt{\frac{C}{B}\log(N)T}$.*

Proof: Note that the algorithm is essentially running the MW algorithm on the estimated losses of the experts. The MW algorithm guarantees (see [2]) that as long as $\hat{\ell}_t(h) \geq 0$ for all t, h, we have for any expert $h^\star$

$$\sum_{t=1}^{T}\sum_h q_t(h)\hat{\ell}_t(h) \;\leq\; \sum_t \hat{\ell}_t(h^\star) + \frac{\eta}{2}\sum_t\sum_h q_t(h)\hat{\ell}_t(h)^2 + \frac{\log N}{\eta}. \tag{2}$$

Now, by Lemma 1 we have

$$\mathbb{E}_t[\sum_h q_t(h)\hat{\ell}_t(h)] \;=\; \sum_h q_t(h)\ell_t(h) \;=\; \mathbb{E}_t[\ell_t(h_t)],$$

and thus $\mathbb{E}[\sum_h q_t(h)\hat{\ell}_t(h)] = \mathbb{E}[\ell_t(h_t)]$. Using this fact, and Lemmas 1 and 2, we see for any h^*,

$$\sum_t \mathbb{E}[\ell_t(h_t)] \;\leq\; \sum_t \ell_t(h^\star) + \frac{\eta}{2}RT + \frac{\log(N)}{\eta}.$$

This gives us the stated regret bound using $\eta = \sqrt{\frac{2\log(N)}{RT}}$ and the fact that $R \leq \frac{2C}{B}$. □

6 Extensions

Re-binning. The analysis of the algorithm given above allows the following flexibility in the algorithm: the algorithm is free to re-bin the experts in each round if it chooses. The regret bound continues to hold as long as the number of bins is bounded by $\frac{2C}{B}$. In practice, this could be very useful if over time it is learned (via side information) that certain groupings of experts are more informative than others. Also, in practice, randomly permuting the experts and then rebinning them yields more stability to the algorithm by avoiding potentially bad binnings. We use this observation in our experiments.

Changing Costs. The algorithm can be extended in a straightforward way to the setting where the costs of the experts and the budget can change with time. Specifically, at time t, the algorithm is told the cost $c_{h,t}$ of querying the advice of expert h, and the budget B_t on the total cost of queried experts. Let $C_t := \sum_{h\in\mathcal{H}} c_{h,t}$ be the total cost of querying all the experts in round t.

In this setting, consider the following variant of the algorithm. In each round t, the experts are re-partitioned into as few bins as possible of total cost at most B_t with the current costs of the experts. The rest of the algorithm stays the same: viz. an expert is chosen from the current probability distribution over the experts, and the bin it belongs to is chosen for querying for expert advice. The update to the distribution and the loss estimators are the same as in Algorithm 1.

It is easy to check that the analysis given above extends to this setting, yielding the following bound:

Theorem 3 *In the setting where in each round t new costs $c_{h,t}$ for the experts and budget B_t are specified, the extension of Algorithm 1 which re-partitions the experts in each round into bins of total cost at most B_t has regret bounded by $O\left(\sqrt{\sum_{t=1}^{T} \frac{C_t}{B_t}\log(N)}\right)$, where $C_t = \sum_{h\in\mathcal{H}} c_{h,t}$.*

Convex Losses. In very common learning settings, the loss of each expert is a convex function of its prediction (for example, in regression settings, one may use the squared loss or absolute loss to measure the error of an expert's prediction). One can obtain lower losses in practice by taking a average of the chosen experts' predictions weighted by their current weight. This does not change the regret bound since Jensen's inequality implies that the loss of the averaged prediction is only lower than the expected loss of a chosen expert.

More precisely, suppose in each round t, we are required to make a prediction $y \in \mathbb{R}$, and the loss of the prediction y is $f_t(y)$ for some convex function $f_t : \mathbb{R} \to [0,1]$. Every expert h makes a prediction $y_t(h) \in \mathbb{R}$, and suffers loss $\ell_t(h) = f_t(y_t(h))$. Then consider a variant of the BEXP algorithm, called BEXP-AVG, which makes the prediction $\bar{y}_t := \frac{\sum_{h\in B_{I_t}} q_t(h) y_t(h)}{\sum_{h\in B_{I_t}} q_t(h)}$. This has lower loss than the expected loss of an expert drawn at random from q_t, by Jensen's inequality: $f_t(\bar{y}_t) \leq \frac{\sum_{h\in B_{I_t}} q_t(h) f_t(y_t(h))}{\sum_{h\in B_{I_t}} q_t(h)}$. Since the probability of choosing the bin B_{I_t} is exactly $r_t(I_t) = \sum_{h\in B_{I_t}} q_t(h)$, we have

$$\mathbb{E}_t[f_t(\bar{y}_t)] \;\leq\; \sum_{i\in\mathcal{R}} \frac{\sum_{h\in B_i} q_t(h) f_t(y_t(h))}{\sum_{h\in B_i} q_t(h)} \cdot \sum_{h\in B_i} q_t(h) \;=\; \sum_{h\in\mathcal{H}} q_t(h) f_t(y_t(h)) = \mathbb{E}_t[\ell_t(h_t)].$$

Using this bound in the proof of Theorem 2, we conclude that this algorithm has the same regret bound. In practice, this variant should have better performance, and indeed in our experiments BEXP-AVG had substantially better performance than other algorithms. It is unclear how to derive similar averaging versions[3] of the previous algorithm of [12].

[3] The natural averaged variant of the algorithm of [12] that makes the weighted prediction $\sum_{h\in S_t} \frac{q_t(h) y_t(h)}{q_t(h)+(1-q_t(h))\frac{B-1}{N-1}}$ does not work. While this weighted prediction is correct in expectation, it does not yield the same regret bound since the prediction is not a convex combination of the observed predictions. This is borne out by experiments as well where it is observed that this averaged variant does not converge.

7 Lower Bound

In this section, we show a lower bound on the regret of any algorithm for the problem which shows that our upper bound is nearly tight.

Theorem 4 *Given the costs c_h for the experts $h \in \mathcal{H}$ and a budget B, for any online algorithm, there is a sequence of expert predictions and losses so that the expected regret of the algorithm is at least $\frac{1}{4}\sqrt{\frac{C}{B}T}$, where $C = \sum_{h \in \mathcal{H}} c_h$.*

Proof: The lower bound is based on the standard information theoretic arguments (see for e.g. [3]). Let $\mathbb{B}(p)$ be the Bernoulli distribution with parameter p, i.e. 1 is chosen with probability p and 0 with probability $1-p$. Let $\mathrm{KL}(\mathbf{P} \parallel \mathbf{Q})$ denote the KL-divergence between distributions $\mathbf{P}$ and $\mathbf{Q}$.

We may assume that $2B \le C$: in the case that $2B \ge C$, an $\Omega(\sqrt{T})$ lower bound on the regret follows directly from the standard $\Omega(\sqrt{T})$ lower bound which holds even in the full-information (i.e. all expert predictions are observed) setting.

In the following, we assume the online algorithm is deterministic (the extension to randomized algorithms is easy by conditioning on the random seed of the algorithm). Fix the parameter $\varepsilon := \frac{1}{2}\sqrt{\frac{C}{BT}}$. The expert predictions and losses are generated randomly as follows. We define N probability distributions, $\mathbf{P}_h$ for all $h \in \mathcal{H}$. Fix $h^\star \in \mathcal{H}$, and we define $\mathbf{P}_{h^\star}$ as follows: in each round t, the loss of expert $h^\star$ is distributed as $\mathbb{B}(\frac{1}{2} - \varepsilon)$, and the losses of all other experts $h \neq h^\star$ are distributed as $\mathbb{B}(\frac{1}{2})$. All random draws are independent of each other. The best expert under this distribution is $h^\star$ with expected loss $\frac{1}{2} - \varepsilon$ in each round. Let $\mathbb{E}_{h^\star}$ denote expectation under $\mathbf{P}_{h^\star}$.

Consider another probability distribution $\mathbf{P}_0$ for the expert predictions and losses where in each round each expert's loss is distributed as $\mathbb{B}(\frac{1}{2})$. Let $\mathbb{E}_0$ denote the expectation of random variables under $\mathbf{P}_0$.

Suppose the expert predictions and losses are generated from $\mathbf{P}_{h^\star}$. Then the algorithm suffers an expected regret of ε whenever $h^\star$ is not in the chosen set of experts, S_t (since the expected loss of any chosen expert is $\frac{1}{2}$ then). Define the random variable $N_{h^\star} = \sum_{t=1}^{T} \mathbf{I}[h^\star \in S_t]$. Then to get a lower bound on the expected regret we need to upper bound $\mathbb{E}_{h^\star}[N_{h^\star}]$. To do this, we use arguments based on KL-divergence between the distributions $\mathbf{P}_{h^\star}$ and $\mathbf{P}_0$. Specifically, for all t, let

$$H_t = \langle (S_1, \ell_1(S_1)), (S_2, \ell_2(S_2)), \ldots, (S_t, \ell_t(S_t)) \rangle$$

denote the history up to time t, where $\ell_\tau(S_\tau) = \langle \ell_\tau(h) \rangle_{h \in S_\tau}$. For convenience, we define $H_0 = \langle\rangle$, the empty vector. Note that since the algorithm is assumed to be deterministic, $N_{h^\star}$ is a deterministic function of the history H_T. Thus to upper bound $\mathbb{E}_{h^\star}[N_{h^\star}]$ we compute an upper bound on $\mathrm{KL}(\mathbf{P}_0(H_T) \parallel \mathbf{P}_{h^\star}(H_T))$. Lemma 3 shows that $\mathrm{KL}(\mathbf{P}_0(H_T) \parallel \mathbf{P}_{h^\star}(H_T)) \le \frac{\varepsilon^2}{2}\mathbb{E}_0[N_{h^\star}]$. Thus, by Pinsker's inequality, we get

$$d_{\mathrm{TV}}(\mathbf{P}_0(H_T), \mathbf{P}_{h^\star}(H_T)) \le \sqrt{\tfrac{1}{2}\mathrm{KL}(\mathbf{P}_0(H_T) \parallel \mathbf{P}_{h^\star}(H_T))} \le \sqrt{\varepsilon^2 \mathbb{E}_0[N_{h^\star}]}.$$

Since $|N_{h^\star}| \le T$, this implies that $\mathbb{E}_{h^\star}[N_{h^\star}] \le \mathbb{E}_0[N_{h^\star}] + \frac{\varepsilon}{2} T \sqrt{\mathbb{E}_0[N_{h^\star}]}$. Consider the distribution over experts where expert h is chosen with probability $\frac{c_h}{C}$. Choosing $h^\star$ from this distribution, taking the expecation under this distribution of both sides of the above inequality, and applying Jensen's inequality to the concave square root function, we get

$$\sum_{h^\star \in \mathcal{H}} \frac{c_{h^\star}}{C} \mathbb{E}_{h^\star}[N_{h^\star}] \;\leq\; \sum_{h^\star \in \mathcal{H}} \frac{c_{h^\star}}{C} \mathbb{E}_0[N_{h^\star}] + \frac{\varepsilon}{2} T \sqrt{\sum_{h^\star \in \mathcal{H}} \frac{c_{h^\star}}{C} \mathbb{E}_0[N_{h^\star}]} \;\leq\; \frac{B}{C} T + \frac{\varepsilon}{2} T^{3/2} \sqrt{\frac{B}{C}}.$$

The second inequality above follows from the following fact:

$$\begin{aligned} \sum_{h^\star \in \mathcal{H}} c_{h^\star} \mathbb{E}_0[N_{h^\star}] &= \sum_{h^\star \in \mathcal{H}} c_{h^\star} \mathbb{E}_0[\sum_t \mathbf{I}[h^\star \in S_t]] \\ &= \sum_t \mathbb{E}_0[\sum_{h^\star \in \mathcal{H}} c_{h^\star} \mathbf{I}[h^\star \in S_t]] \\ &\leq \sum_t B \;=\; BT, \end{aligned}$$

where the inequality follows because the cost of the experts in S_t does not exceed B.

Since the expected regret when the distribution of expert predictions and losses is $\mathbf{P}_{h^\star}$ is at least $\varepsilon(T - \mathbb{E}_{h^\star}[N_{h^\star}])$, and $\frac{B}{C} \leq \frac{1}{2}$, we have

$$\sum_{h^\star \in \mathcal{H}} \frac{c_{h^\star}}{C} \mathbb{E}_{h^\star}[\text{Regret}] \;\geq\; \sum_{h^\star \in \mathcal{H}} \frac{c_{h^\star}}{C} \varepsilon(T - \mathbb{E}_{h^\star}[N_{h^\star}]) \;\geq\; \frac{\varepsilon}{2} T - \frac{\varepsilon^2}{2} T^{3/2} \sqrt{\frac{B}{C}} \;=\; \frac{1}{4} \sqrt{\frac{B}{C} T},$$

using the setting $\varepsilon = \frac{1}{2}\sqrt{\frac{C}{BT}}$. Thus there exists some $h^\star$ such that $\mathbb{E}_{h^\star}[\text{Regret}] \geq \frac{1}{4}\sqrt{\frac{B}{C} T}$. $\square$

The following lemma gives an upper bound on the KL-divergence between $\mathbf{P}_0(H_T)$ and $\mathbf{P}_{h^\star}(H_T)$.

Lemma 3 *$KL(\mathbf{P}_0(H_T) \parallel \mathbf{P}_{h^\star}(H_T)) \;\leq\; \frac{\varepsilon^2}{2} \mathbb{E}_0[N_{h^\star}]$.*

PROOF: For notational brevity, let $\mathbf{P}_0^t$ and $\mathbf{P}_{h^\star}^t$ denote probability distributions conditioned on the choice of H_{t-1}. We have

$$\begin{aligned} \mathrm{KL}(\mathbf{P}_0(H_T) \parallel \mathbf{P}_{h^\star}(H_T)) &= \sum_{t=1}^T \mathrm{KL}(\mathbf{P}_0^t(S_t, \ell_t(S_t)) \parallel \mathbf{P}_{h^\star}^t(S_t, \ell_t(S_t))) && (3) \\ &= \sum_{t=1}^T [\mathrm{KL}(\mathbf{P}_0^t(\ell_t(S_t)|S_t) \parallel \mathbf{P}_{h^\star}^t(\ell_t(S_t)|S_t)) + \mathrm{KL}(\mathbf{P}_0^t(S_t) \parallel \mathbf{P}_{h^\star}^t(S_t))] && (4) \\ &= \sum_{t=1}^T \mathrm{KL}(\mathbf{P}_0^t(\ell_t(S_t)|S_t) \parallel \mathbf{P}_{h^\star}^t(\ell_t(S_t)|S_t)) && (5) \\ &= \sum_{t=1}^T \mathbf{P}_0[h^\star \in S_t] \mathrm{KL}(\mathbb{B}(\tfrac{1}{2}) \parallel \mathbb{B}(\tfrac{1}{2} - \varepsilon)) && (6) \\ &\leq \sum_{t=1}^T \mathbf{P}_0[h^\star \in S_t] \cdot \frac{\varepsilon^2}{2} \;\leq\; \frac{\varepsilon^2}{2} \mathbb{E}_0[N_{h^\star}]. && (7) \end{aligned}$$

Equalities (3) and (4) follow from the chain rule for relative entropy. Equality (5) follows because the distribution of S_t conditioned on H_{t-1} is identical in $\mathbf{P}_0$ and $\mathbf{P}_{h^\star}$. Equality (6) needs some

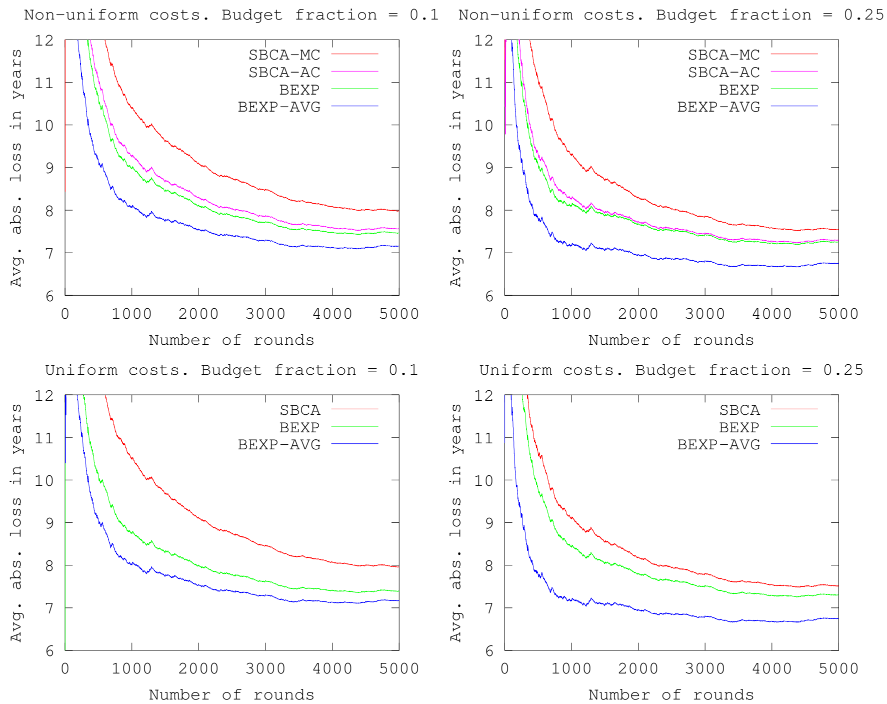

Figure 1: Comparison of the average loss of different budgeted learning algorithms for different budget levels (expressed as a fraction of the total cost of querying all experts). Top row: non-uniform costs for experts. Bottom row: uniform costs for experts.

explanation. If $h^\star \notin S_t$, then the losses $\ell_t(h)$ for all $h \in S_t$ are distributed as $\mathbb{B}(\frac{1}{2})$ under both under both $\mathbf{P}_0$ and $\mathbf{P}_{h^\star}$, and so the KL-divergence is 0. If $h^\star \in S_t$, then the distribution of $\ell_t(h)$ for all $h \neq S_t$ is $\mathbb{B}(\frac{1}{2})$ under both $\mathbf{P}_0$ and $\mathbf{P}_{h^\star}$, whereas the distribution of $\ell_t(h^\star)$ is $\mathbb{B}(\frac{1}{2})$ under $\mathbf{P}_0$ and $\mathbb{B}(\frac{1}{2} - \varepsilon)$ under $\mathbf{P}_{h^\star}$. Since the losses for all experts are independently chosen, we conclude that if $h^\star \in S_t$, the KL-divergence is exactly $\mathrm{KL}(\mathbb{B}(\frac{1}{2}) \parallel \mathbb{B}(\frac{1}{2} - \varepsilon))$. Finally, inequality (7) follows using standard calculations for KL-divergence between Bernoulli random variables. $\square$

8 Experiments

The following experiments demonstrate the algorithm of section 4 and how it might be used for model selection in a resource-constrained system making online predictions. We will use the year-prediction task associated with the Million Song Dataset (MSD). Each example in the MSD corresponds to a song released between 1922 and 2011. We use the same features as [6] – 90 acoustical features representing the timbre of a song. The goal is to predict the year in which a song was released.

As discussed in the introduction, we think of each expert as being a model trained using a variety of software, as is common in real systems. We generate a variety of experts by taking a particular parameterizable learning algorithm and generating multiple instances, each with a different combination of parameter settings. In this way, we can think of finding the best expert as corresponding to online parameter tuning.

8.1 Experimental setup

We used 3 families of models trained with different parameter settings, for a total of 29 different experts. The models were all trained on a set of $46,382$ examples of the MSD. All labels in the dataset were in the range 1929 to 2010. We normalized the labels by subtracting 1900 and dividing by 100 so that they were in the range $[0, 1.1]$.

In more detail, the models used are as follows:

1. k-Nearest Neighbor trained using MATLAB. We used k values in $\{1, 5, 11, 16, 21, 26, 51\}$, and either uniform averaging or inverse distance weighted averaging to compute the label, giving rise to fourteen models.

2. SVM models trained using libsvm[4]. We trained a total of nine different models corresponding to using either ϵ-SVR or ν-SVR, with a variety of kernels (linear, polynomial, and sigmoid).

3. Linear regression models trained using Vowpal Wabbit[5]. We trained six different models using either 1, 2, 5, or 10 passes over the training data, with additional variation due to adding an optional L-BFGS step and scaling the learning rate.

Cost for each model was measured in terms of running time for each example. Within each model family, the cost is roughly the same for different parameter settings. Thus, after normalizing time, the costs for each SVM, k-Nearest Neighbor and linear regression model were 147, 17 and 2 time units per example, respectively.

For testing, we applied the same normalization applied to labels to the predictions of the experts. Furthermore, any normalized predictions of experts outside the range $[0, 1.1]$ were clamped to this range. We used absolute loss to measure the performance of the experts, i.e., if on example (x, y), an expert predicted $\hat{y}$, the loss is $|y - \hat{y}|$. We then tested the performance of the budgeted prediction algorithms developed in this paper on a sequence of $T = 5,153$ examples arriving online. On each round t, given an example (x_t, y_t), the algorithm chooses a subset of experts to query, obtains their predictions, makes its own prediction $\hat{y}_t$, and suffers the absolute loss $|y_t - \hat{y}_t|$.

We compared the performance of our algorithms to the algorithm of [12] which works in the uniform costs setting. Their algorithm (dubbed SBCA) operates by sampling an expert from a distribution, using its prediction. Additional experts are then sampled uniformly at random. The distribution is updated using exponential weighting of the appropriately constructed unbiased loss estimators. Since their algorithm only works in the uniform costs case, to adapt to the non-uniform costs case considered in this paper, we set all expert costs to be some fixed cost c, and set the number of experts to be queried in their algorithm to be $\lfloor \frac{B}{c} \rfloor$. Setting c equal to the maximum cost of any expert ensures that the budget is never exceeded. We call the resulting algorithm SBCA-MC. Another option is to set c to be the average cost of all experts. However in this case it is possible that the budget is exceeded in some round. In this case, we simply stop sampling experts as soon as the budget limit is reached. We call the resulting algorithm SBCA-AC. While this algorithm generates biased estimates of the true losses and is hence not guaranteed to converge, in experiments it did seem to converge.

[4] `http://www.csie.ntu.edu.tw/~cjlin/libsvm/`
[5] `http://hunch.net/~vw/`

8.2 Experimental Results

In the first set of experiments, we tested the performance of the algorithms with the same dataset with different budget levels, using the costs of the experts as specified above. These budget levels are expressed as a fraction of the total cost of querying all experts; a fraction of 1 corresponds to being able to query all experts. We plot the average absolute loss in years, averaged over 10 runs of the algorithm, for budget levels 0.1 and 0.25 in Figure 1. The top row of the figure shows the results for the non-uniform cost setting. We observed that BEXP-AVG is substantially better than the other algorithms because it levarges the convexity of the loss function. While BEXP is competitive with SBCA-AC, we note that SBCA-AC doesn't have a proof of convergence. As expected, SBCA-MC has the worst performance. Similar performance was observed on other datasets as well.

In the second set of experiments, to obtain a more fair comparison to SBCA, we also ran experiments with the same dataset except that the costs of all experts were now set to 1, so that we are in the uniform costs setting. In this case as well, the results are similar: again BEXP-AVG is substantially better than the others, and BEXP outperform SBCA at low budget levels, and is competitive at higher budget levels.

9 Conclusion

In this work we considered the problem of learning from expert advice, where there is a budget constraint on the total cost of experts that can be queried on a prediction round. We give an algorithm, BEXP, which attains nearly optimal rates, and interpolates between the standard experts and multiarmed bandits settings.

While a special case of this setting — wherein experts have uniform costs — has been considered by [12], it's unclear how to adapt that algorithm to the generalized cost setting. We argue the importance of the generalized cost setting. In particular, we demonstrate how BEXP can be used for online model selection and parameter tuning. Finally, we emphasize the flexibility of our algorithm, which easily accommodates changing costs, re-binning of experts, and averaging under convex losses. While rebinning and averaging do not improve theoretical rates, they are extremely useful in practice. This allows us to empirically outperform [12] even in the uniform cost setting.

References

[1] Noga Alon, Nicolò Cesa-Bianchi, Claudio Gentile, and Yishay Mansour. From bandits to experts: A tale of domination and independence. In *NIPS*, pages 1610–1618, 2013.

[2] Sanjeev Arora, Elad Hazan, and Satyen Kale. The Multiplicative Weights Update Method: a Meta-Algorithm and Applications. *Theory of Computing*, 8(1):121–164, 2012.

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

[4] Ashwinkumar Badanidiyuru, Robert Kleinberg, and Aleksandrs Slivkins. Bandits with knapsacks. In *FOCS*, pages 207–216, 2013.

[5] Shai Ben-David and Eli Dichterman. Learning with restricted focus of attention. In *COLT*, pages 287–296. ACM, 1993.

[6] Thierry Bertin-Mahieux, Daniel P.W. Ellis, Brian Whitman, and Paul Lamere. The million song dataset. In *ISMIR*, 2011.

[7] Nicolò Cesa-Bianchi, Yoav Freund, David Haussler, David P. Helmbold, Robert E. Schapire, and Manfred K. Warmuth. How to use expert advice. *J. ACM*, 44(3):427–485, 1997.

[8] Nicolo Cesa-Bianchi, Shai Shalev-Shwartz, and Ohad Shamir. Efficient learning with partially observed attributes. *The Journal of Machine Learning Research*, 12:2857–2878, 2011.

[9] Helmut Grabner, Michael Grabner, and Horst Bischof. Real-time tracking via on-line boosting. In *British Machine Vision Conference*, pages 47–56, 2006.

[10] Elad Hazan and Tomer Koren. Linear regression with limited observation. In *ICML*, 2012.

[11] Satyen Kale. Multiarmed bandits with limited expert advice. In *COLT*, 2014.

[12] Yevgeny Seldin, Peter L. Bartlett, Koby Crammer, and Yasin Abbasi-Yadkori. Prediction with limited advice and multiarmed bandits with paid observations. In *ICML*, pages 280–287, 2014.

[13] Long Tran-Thanh, Archie C Chapman, Alex Rogers, and Nicholas R Jennings. Knapsack based optimal policies for budget-limited multi-armed bandits. In *AAAI*, 2012.

[14] Vijay V. Vazirani. *Approximation algorithms.* Springer, 2001. ISBN 978-3-540-65367-7.

[15] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang. Online object tracking: A benchmark. In *Computer Vision and Pattern Recognition*, pages 2411–2418, 2013.

[16] Navid Zolghadr, Gábor Bartók, Russell Greiner, András György, and Csaba Szepesvári. Online learning with costly features and labels. In *NIPS*, pages 1241–1249, 2013.