Second-Order Bounds for Sleeping Experts

Noah Bergam njb2154@columbia.edu
Arman Özcan ao2794@columbia.edu
Daniel Hsu djhsu@columbia.edu
Department of Computer Science
Columbia University
New York, NY 10027, USA

Editor:

Abstract

In this report we discuss the *sleeping experts problem*, a variant of online allocation where only a subset of the experts $E_t \subset [N]$ are available at each time step t. We compare the two main benchmarks for regret bounds: one based on the best ranking of experts, and one based on the best mixture of experts. We prove these benchmarks are equivalent in general, and the proof illuminates a fundamental connection between sleeping experts and Plackett-Luce models, a well-studied statistical model for rankings. We also demonstrate that the best-ordering loss is NP-hard to compute. Furthermore, we develop the first beyond-worst-case regret bounds for sleeping experts, replacing the dependence on the number of rounds with the cumulative variance of the losses over T rounds.

Keywords: online allocation, sleeping experts, second-order bounds

1 Introduction

Learning with experts and multi-armed bandits are prototypical online learning problems. The key difference between the two lies in the information revealed to the learner: in the former, the learner sees the loss of every action, whereas in multi-armed bandits, the learner sees only the loss of the chosen action. Against an oblivious adversary, the former can be solved with $O(\sqrt{T \log N})$ regret by the Hedge algorithm (Littlestone and Warmuth, 1994). The latter can be solved with $O(\sqrt{TN \log N})$ regret by EXP3. Both algorithms are random and hence the regret bounds are in expectation.

In this report, we study the sleeping experts problem, another partial-information variant of learning with experts. The complication here is that every day, only a subset $E_t \subset [N]$ of experts are observed and available to play. It is not immediately obvious how to define a regret benchmark in this set-up: comparing experts directly would be inappropriate because some may play for many more rounds than others. Comparing mixtures of experts can work, but it is crucial that the loss incurred in each round is normalized by the mass of awake experts. This gives rise to our first notion of regret for sleeping experts:

Definition 1 *The* ***distributional loss*** *of a sleeping experts sequence* $\{(l_t, E_t)\}_{t\in[T]}$ *is*

$$L_{T,u} = L_T(u) = \sum_{t=1}^{T} \frac{1}{u(E_t)} \sum_{i\in E_t} u_i l_{t,i}$$

where p_t is the probability vector played by the learner at round t and $u \in \Delta_{N-1}$.

The second way, introduced by (Kleinberg et al., 2010), is to measure performance in terms of a ranking of available actions. Note that, if the manner of choosing an action is random, the following regret bound is random and hence could be analyzed in expectation.

Definition 2 *The* ***rank loss*** *of a sleeping experts sequence $\{(l_t, E_t)\}_{t\in[T]}$ is*

$$L_{T,\sigma} = L_T(\sigma) = \sum_{t=1}^{T} l_{t,\sigma(E_t)}$$

where $\sigma(E_t) = \min_{i\in E_t} \sigma(i)$ is the top-ranked awake expert according to some fixed permutation $\sigma \in S_N$, and i_t is the action chosen by the learner at round t.

Compare these notions to the usual notion of loss for online allocation, $L_{T,i} = \sum_{t=1}^{T} l_{t,i}$. The main contributions of this paper are as follows:

- We compare the distributional and rank benchmarks in novel detail. We show explicit examples of the distributional benchmark not being well-defined as a minimum. Furthermore, we find that the distribution and rank benchmark coincide in expectation when either availability of experts or losses is stochastic. We conjecture this equivalence $\inf_u L_T(u) = \min_\sigma L_T(\sigma)$ hold in the general case, where both expert availability and loss is adverserial.
- We show, via a straightforward reduction from minimum feedback arc set, that computing the optimal rank benchmark is an NP-hard problem.
- We present novel second-order bounds for two different settings of sleeping experts. We derive a $O(\sqrt{\text{VAR}_T^{\max} \log N})$ bound for sleeping experts in a stochastic availability, Adversarial loss framework, and a $O(\sqrt{\text{VAR}_T^{\max} N \log N})$ bound for the fully Adversarial framework. $\text{VAR}_T^{\max}$ is a measure of the cumulative variance of the loss vectors, defined formally in Section 4.

2 Background

In this section, we review the basic definitions and concepts behind the sleeping experts problem; the major results and open questions; and the applications in prediction problems.

2.1 Definitions

Discrete Sleeping Experts Suppose there are N experts making binary predictions every day: specifically, expert i on day t predicts $x_{t,i} \in \{0, 1\}$. Assume that predictions and true outcomes are from $[0, 1]$. We define some non-negative loss measure $L : [0, 1]^2 \to [0, \infty)$ that takes in a prediction and an outcome and outputs a measure of how different they are. The sleeping experts game proceeds as follows.

- Adversary chooses the set of awake experts $E_t \subseteq [N]$ and predictions $x_{t,i}$ for each awake expert $i \in E_t$ at time t.

- Learner sees $\{x_{t,i} : i \in E_t\}$ and makes a prediction $\hat{y}_t$.
- Adversary chooses outcome y_t.
- Learner incurs loss $L(\hat{y}_t, y_t)$ and each awake expert i incurs loss $L(x_{t,i}, y_t)$.

The goal is to make the total loss of learner, $\sum_{t=1}^{T} L(\hat{y}_t, y_t)$, as small as possible compared to the total loss of the best fixed mixture of the "awake" experts, which can be defined as:

$$\min_{\mathbf{u} \in \Delta_N} \sum_{t=1}^{T} \frac{\sum_{i \in E_i} u_i L(x_{t,i}, y_t)}{\sum_{i \in E_i} u_i}$$

which computes the weighted average loss of the awake experts according to some best-in-hindsight fixed distribution $\mathbf{u} \in \Delta_N$. Alternatively, we can define the best fixed mixture as the loss of the weighted average prediction according to best $\mathbf{u} \in \Delta_N$ as:

$$\min_{\mathbf{u} \in \Delta_N} L\left(\frac{\sum_{i \in E_i} u_i x_{t,i}}{\sum_{i \in E_i} u_i}\right)$$

Note that if the loss is convex, a bound on the second implies a bound on the first. Regardless of which definition above is chosen, denote the loss of fixed mixture of the "awake" experts at time t with respect to $\mathbf{u}$ as $L_{\mathbf{u}}(x_t, y_t)$.

(Freund et al., 1997) provided a general method to convert an insomniac algorithm (where every expert is always awake) and its respective regret bound into a corresponding sleeping algorithm (where some experts sometimes sleep) and regret bound. Both algorithms maintain a probability distribution vector $p_t \in \Delta_N$ as weights for N experts and define two general black-box functions $\mathsf{predict}_N$ and update_N: The function $\mathsf{predict}_N$ maps the set of N expert predictions x_t and weights p_t to a prediction $\hat{y}_t$. The function update_N takes in the set of N expert predictions x_t and weights p_t and the true outcome y_t and outputs an updated set of expert weights p_{t+1}. The subscript N is dropped for notational simplicity. The general form of an insomniac algorithm is as follows:

- See the predictions of experts x_t and predict $\hat{y}_t = \mathsf{predict}(x_t, p_t)$ at time t.
- See true outcome y_t and incur loss $L(\hat{y}_t, y_t)$.
- Update the weights as $p_{t+1} = \mathsf{update}(x_t, p_t, y_t)$.

which is converted into a sleeping setting as:

- See the predictions of awake experts $x_t^{E_t}$ and predict $\hat{y}_t = \mathsf{predict}(x_t^{E_t}, p_t^{E_t})$ at time t.
- See true outcome y_t and incur loss $L(\hat{y}_t, y_t)$.
- Update the weights of awake experts as $p_{t+1}^{E_t} = \mathsf{update}(x_t^{E_t}, p_t^{E_t}, y_t)$ while keeping the weights of sleeping experts the same and maintaining $\sum_{i=1}^{N} p_{t+1,i} = 1$.

where $x_t^{E_t}$ is the set of predictions of awake experts at time t and $p_t^{E_t}$ is the set of normalized weights of awake experts, such that $p_{t,i}^{E_t} = \frac{p_{t,i}}{\sum_{i \in E_t} p_{t,i}}$ for all $i \in E_t$.

Sleeping Expert Allocation We generalize sleeping experts in an analogous manner to the way the basic learning with experts problem becomes online allocation. The *sleeping expert allocation game (full information)* proceeds as follows:

- Every day $t = 1, ..., T$:
  1. Learner commits to an allocation vector $p_t \in \Delta_{N-1}$.
  2. Adversary reveals $E_t \subset [N]$.
  3. Learner samples $i_t \sim p_t^{E_t}$ and incurs loss l_{t,i_t}.
  4. Adversary reveals $l_{t,i}$ for **all** $i \in E_t$.

We can reduce to the discrete sleeping experts setting with $l_{t,i} = L(x_{t,i}, y_t) \cdot \mathbf{1}(i \in E_t)$ and $\mathsf{predict}(x_t^{E_t}, p_t^{E_t}) = i_t$ sampled from $p_t^{E_t}$ (note that, in this reduction, we do not use today's expert predictions at all). We often care about the expected performance of an algorithm on the sleeping allocation game, which is given by:

$$\mathbb{E}[l_{t,i_t}] = \sum_{i \in E_t} p_{t,i}^{E_t} L(x_{t,i}, y_t) = \sum_{i \in E_t} \frac{p_{t,i}}{\sum_{i \in E_t} p_{t,i}} L(x_{t,i}, y_t) = \frac{\sum_{i \in [N]} p_{t,i} l_{t,i}}{\sum_{i \in E_t} p_{t,i}} = \frac{\langle p_t, \mathbf{l}_t \rangle}{\sum_{i \in E_t} p_{t,i}}$$

Note that, when summed over t, this is very similar to the distribution benchmark, the key difference being that the algorithm plays a time-varying distribution, whereas the distribution benchmark is defined according to a fixed distribution.

Furthermore, observe that in the insomniac setting, $\mathbb{E}[L(\hat{y}_t, y_t)] = \langle p_t, \mathbf{l}_t \rangle$.

Variations Note that we can also consider a *partial information* ("bandits") setting of sleepy online allocation. The only difference is that we remove step 4, and the learner only gets to see the loss of the chosen action l_{t,i_t}. In either the full or partial information case of sleeping experts, we are optimizing the following notion of regret:

$$R_T = \mathbb{E}_{\mathsf{alg}} \Big[\sum_{t=1}^{T} l_{t,i_t} \Big] - \min_{(*)} L_{T,(*)} \tag{1}$$

where $(*)$ depends on the choice of distribution or rank benchmark and the expectation on the first term is taken with respect to the randomness in the algorithm.

Following the work of (Kleinberg et al., 2010) and (Kanade et al., 2009), we consider cases where the adversary is restricted to acting according to a fixed distribution, either when choosing losses or availability of experts.

- **Stochastic Availability**: $E_t \sim \mathbb{P}_{\text{avail}}$, a fixed distribution over $\mathcal{P}([N])$, every day t.
- **Stochastic Loss**: Every day t, $l_{t,i} \sim \mathbb{P}_{t,i}$ with time-invariant mean $\mu_{t,i} = \mu_i$ for each $i \in E_t$.

We then usually consider regret in some sort of expectation over these distributions. Note that this may change the notion of benchmarks. For instance, our second-order regret bound stochastic availability compares performance to $\min_\sigma \mathbb{E}_{E_t} L_{T,\sigma}$ rather than the expectation on the outside.

2.2 Results for Sleeping Experts

(Freund et al., 1997) introduced the sleeping experts problem, and, as discussed in the previous section, proposed a general-purpose framework for converting an insomniac online learning algorithm into a one that can handle the sleeping case. They also showed that, for a log loss $L(\hat{y}, y) = -\ln(\hat{y})\mathbf{1}(y = 1) - \ln(1 - \hat{y})\mathbf{1}(y = 0)$, a sleeping version of the Bayes algorithm achieves:

$$\mathbb{E}_{\mathsf{alg}} \sum_{t=1}^{T} u(E_t) l_{t,i_t} - \sum_{t=1}^{T} u_i l_{t,i} \leq \mathsf{KL}(u||p_1)$$

where $l_{t,i} = L(x_{t,i}, y_t)$ and we use the fact that the log loss is convex (see the original statement in Theorem 1, Freund et al. (1997)). Ultimately, if we re-weight $l_{t,i} \to l_{t,i}/u(E_t)$ and set p_1 to a uniform distribution, we achieve $R_T \leq \ln(N)$ (for R_T defined in 1, with comparison to the distribution benchmark).

(Blum and Mansour, 2007) discuss a generalization of sleeping experts where, in effect, experts can be partially awake. More precisely, they discuss a notion of external regret where we consider performance against a collection of *time-selection functions* $I : [T] \to [0, 1]$. The regret of an algorithm alg compared to an expert $i \in [N]$ with respect to time-selection function I is given by:

$$R_{i,I}^{\mathsf{alg}} = \sum_{t=1}^{T} I(t) \Big[l_{t,\mathsf{alg}(t)} - l_{t,i} \Big] = L_{\mathsf{alg},I} - L_{i,I}$$

The sleeping experts problem can be seen as a specialization of this in the following sense: consider pairs $(u, i, I_{u,i})$ for each $i \in [N]$ and $u \in \Delta_{N-1}$, where $I_{u,i}(t) = \frac{u_i}{u(E_t)}\mathbf{1}(i \in E_t)$. Then we recover the distributional regret as follows:

$$\sum_{i=1}^{N} R_{i,I_{u,i}}^{\mathsf{alg}} = \sum_{i=1}^{N} L_{\mathsf{alg},I_{u,i}} - L_{i,I_{u,i}} = \sum_{t=1}^{T} \sum_{i \in E_t} \frac{u_i}{u(E_t)} (l_{t,\mathsf{alg}(t)} - l_{t,i}) = \sum_{t=1}^{T} l_{t,\mathsf{alg}(t)} - \underbrace{\sum_{t=1}^{T} \sum_{i \in E_t} \frac{u_i}{u(E_t)} l_{t,i}}_{L_{t,u}}$$

(Kleinberg et al., 2010) introduced the rank benchmark for sleeping experts and sketched how the time-selection model of (Blum and Mansour, 2007) can also reduce to it. Namely, define triples $(\sigma, i, I_{\sigma,i})$ where $I_{\sigma,i}(t) = \mathbf{1}(i = \sigma(E_t))$. Then we have:

$$\sum_{i=1}^{N} R_{i,I_{\sigma,i}}^{\mathsf{alg}} = \sum_{t=1}^{T} \sum_{i:i=E_t} (l_{t,\mathsf{alg}(t)} - l_{t,i}) = \sum_{t=1} l_{t,\mathsf{alg}(t)} - \underbrace{\sum_{t=1}^{T} l_{t,\sigma(E_t)}}_{L_{t,\sigma}}$$

Note that the sum over i disappears since exactly one $i = \sigma(E_t)$ for each round t.

These vignettes suggest that the time-selection framework is quite general. However, that generality comes at a cost; the regret bounds obtained by (Blum and Mansour, 2007) are sub-optimal, both in terms of computational runtime and the information-theoretically necessary number of rounds. (Kleinberg et al., 2010) points out that directly applying their guarantee that

$$\max_{I \in \mathcal{I}} \min_{i \in [N]} R_{i,I} \leq \sqrt{(\max_{I} \min_{i} L_{I,i}) \log(N \cdot |\mathcal{I}|)} + \log(N \cdot |\mathcal{I}|)$$

gives regret bound $O(\sqrt{Tn^2 \log n} + n \log n)$ for the rank regret (and does not apply to the distribution settings, since $|\{I_u : u \in \Delta_{N-1}\}|$ is infinite).

(Kleinberg et al., 2010) improves on this bound, showing that an application of Hedge on $N!$ experts (one for each permutation) achieves $O(\sqrt{TN \log N})$. In the partial information setting, EXP4 with $N!$ experts and N actions achieves $O(N\sqrt{T \log N})$. Both are information-theoretically optimal (in the sense that they prove matching lower bounds). The difficulty is that the algorithms are not computationally efficient, as they require the maintenance of $N!$ weights.

(Kleinberg et al., 2010) also develops bounds for the stochastic loss. In the full-information case, they have a lower bound of $\Omega(\sum_{i=1}^{n}(\mu_i - \mu_{i+1})^{-1})$ where the μ_i means of all the experts are listed in descending order. This lower bound is a achieved by an algorithm they dub Follow the Awake Leader (FTAL).(Kanade et al., 2009) extends this work by studying the sleeping experts model under stochastic action availability. Most notably, they find an efficient algorithm for stochastic availability and Adversarial losses, which we further extend in this paper (to accomodate second-order bounds).

2.3 Results for Second-Order Bounds

Recall that the standard Hedge bound, from (Littlestone and Warmuth, 1994), is given by

$$R_{T,k} = \sum_{t=1}^{T} \langle p_t, l_t \rangle - \sum_{t=1}^{T} l_{t,k} \leq \frac{\ln N}{\eta} + \eta \sum_{t=1}^{T} l_{t,k}^2$$

If we assume $l_{t,k} \in [0,1]$, we can bound the rightmost term by T and then tune η using the doubling trick to arrive at a $O(\sqrt{T \log N})$ bound. If we had hindight knowledge of the losses, we could get a $O(\sqrt{(\sum_{t=1}^{T} l_{k,t}^2) \log N})$ regret bound, which would be nice since it adapts to the hardness of the problem (if the losses are largely small, we expect to converge faster). Unfortunately, as explained by (Gaillard et al., 2014), standard tuning methods like the doubling trick cannot immediately fix this issue (since the optimal η would depend on a non-monotone sequence).

There are, however, workarounds that allow us to replace T with some sort of *second-order statistic*. (Hazan and Kale, 2010) had one of the first results in this direction, replacing T with the maximum cumulative variance of loss for the optimal expert, $\text{VAR}_T^{\max}$: see Section 4, Equation 12 for the precise definition. This bound is uniform over all experts. (Gaillard et al., 2014) replaces this with an expert-dependent second-order bound:

$$R_{T,k} \lesssim \sqrt{\ln N \sum_{t=1}^{T} \Big(\langle p_t, l_t \rangle - l_{k,t} \Big)^2} \qquad \forall k \in [N]$$

Finally, there are other notions of beyond worst case bounds for online allocation, which, loosely speaking, address the $\log N$ rather than the T in the Hedge bound. Let p_1 be the initial setting of weights for online allocation (the prior). *Quantile bounds* like that of (Chaudhuri et al., 2009) isolate K such that:

$$\min_{k \in K} R_{T,k} \lesssim \sqrt{T \ln(1/p_1(K))}$$

Note that this reduces to the original bound if our prior p_1 is uniform and $|K| = 1$. The hope is that K is a large set, such that we can guarantee that it's worst member does better than the $\ln N$ bound. (Koolen and Van Erven, 2015) presents an algorithm that combines second-order and quantile methods.

3 Relating the Notions of Regret

In this section, we compare two notions of cumulative loss. We find that the optimization of the distribution loss is not straightforward.

Definition 3 *Define the best distributional and rank benchmark, respectively, as:*

$$L_T^{(dist)} = \inf_{u \in \Delta_{N-1}} L_T(u) \quad \textit{and} \quad L_T^{(rank)} = \min_{\sigma \in S_N} L_T(\sigma)$$

3.1 On Computing the Benchmarks

In this section, we consider the problem of actually computing the distribution and rank benchmarks, assuming offline access to the loss vectors. Our first observation is that minimizing $L_T(u)$ is a non-convex, ruling out guarantees for gradient-based methods.

Claim 1 *There exists $l_1, .., l_T$ such that the distribution loss $L_T(u) = L_T(u; l_1, ..., l_T)$ is non-convex in u.*

Proof Consider the following loss table:

	Expert 1	**Expert 2**	**Expert 3**
Round 1	asleep	0	1
Round 2	0	1	asleep
Round 3	1	asleep	0

Table 1: Non-convexity of distribution loss with respect to u.

Consider the weight vectors $u_1 = (1 - \epsilon, \epsilon, 0)$ and $u_2 = (\epsilon, 0, 1 - \epsilon)$ for arbitrarily small $\epsilon > 0$. Then,

$$L_T(u_1) = 0 + \epsilon + 1 = 1 + \epsilon \quad \text{and} \quad L_T(u_2) = 1 + 0 + \epsilon = 1 + \epsilon$$

Take the average of two weight vectors: $u' = \frac{u_1 + u_2}{2} = (1, \epsilon, 1 - \epsilon)\frac{1}{2}$. Then,

$$L_T(u') = (1 - \epsilon) + \frac{\epsilon}{1 + \epsilon} + \frac{1}{2 - \epsilon}$$

For example, for $\epsilon = 1/4$, $L_T(u') \approx 1.52$ but $L_T(u_1) = L_T(u_2) = 1.25$. In fact, as ϵ approaches 0, $L_T(u') \to 1.5$ but $L_T(u_1) = L_T(u_2) \to 1$. Hence, $L_T(u)$ is not convex. ■

Note that above example also proves that the best ranking of experts is not unique. The rankings $\sigma_1 = (1 - 2 - 3)$ and $\sigma_2 = (3 - 1 - 2)$ both minimize the rank loss and give $L_T(\sigma_1) = L_T(\sigma_2) = 1$.

While computing the distribution benchmark is a continuous optimization problem, the rank benchmark is a combinatorial problem. A brute force solution takes superexponential time $O(N!)$. If $E_t = [N]$ every day (insomniac setting) then the best ranking is easily computed as $\sigma^* = \text{sort}_i(\sum_{t=1}^{T} l_{t,i})$ in increasing order. Also, as shown in sections below, if either losses or expert availabilities are stochastically chosen, it is again easy to compute the best ranking. It turns out that best ranking just amounts to sorting losses, either the average losses or the total losses across all time steps, depending on whether the losses are stochastic or not.

For the case when E_t and l_t are both adversarial, however, there is no efficient algorithm to our knowledge that finds the best ranking. Naive approaches such as ordering according to the total losses incurred by each expert or the total number of time steps each expert "wins" (has the smallest loss) do not give the best ranking. It is not even true that the best expert provides the smallest total loss in the time steps that it is awake at. Consider the following setup:

	Expert 1	**Expert 2**	**Expert 3**
Round 1	asleep	1	100
Round 2	1	100	0
Round 3	1	10	0

Table 2: Demonstrating that the best expert need not provide the smallest total loss in the time steps where it is awake.

For the best ranking, Expert 3 must be behind Expert 2 and Expert 2 behind Expert 1, which leads to the best ranking of $\sigma^* = 1-2-3$, which suffers the rank loss $L_T(\sigma^*) = 3$. However, note that Expert 1 suffers total loss of 2 in the rounds that it is awake, whereas Expert 2 suffers 0 for the same rounds.

Indeed, we find that calculating the best ranking is NP-hard in general. We state the problem below.

- **Best-Ordering for Sleeping Experts (BOSE)**: Given loss vectors $l_1, ..., l_T \in \mathbb{R}_+^N$, expert availabilities $E_1, ..., E_T \subset [N]$, and $L > 0$, compute $\sigma^* = \text{argmin}_{\sigma \in S_N} \sum_{t=1}^{T} l_{t,\sigma(E_t)}$.

Note that, using binary search, we can use an oracle for this problem to find the loss of the optimal σ. The problem is clearly in NP, where σ is the certificate. However, if BOSE is an NP-hard problem, computing the best rank loss is clearly also NP-hard, as is finding the optimal ranking.

Theorem 1 *BOSE is an NP-complete search problem.*

Proof It is clear that BOSE is in NP (σ^* being the certificate for the corresponding decision problem). To show NP-hardness, we proceed via a reduction from minimum feedback arc set (MFAS for short), a well-known NP-hard problem. Start with an instance of $G = (V = [n], E)$. Let the experts correspond to vertices, and the rounds correspond to directed edges.

For each round $e = (i \to j) \in E$, the awake experts are the adjacent vertices i and j (i.e. $E_u = \{i, j\}$). The loss vector is as follows:

$$\text{for } e = (i \to j) \qquad l_{e,i} = 0 \qquad l_{e,j} = 1$$

So the BOSE loss becomes:

$$\begin{aligned}\sum_{(i\to j)\in E} l_{e,\sigma(\{i,j\})} = \sum_{(i\to j)\in E} \mathbf{1}[j = \sigma(\{i,j\})] &= \sum_{(v_i\to v_j)\in E} \mathbf{1}[j = \operatorname*{argmin}_{j'\in\{i,j\}} \sigma(j')] \\ &= \sum_{(v_i\to v_j)\in E} \mathbf{1}[\sigma(j) < \sigma(i)]\end{aligned}$$

On the RHS, we recover exactly the objective function for MFAS. Hence, a black-box poly-time solver for BOSE gives us a poly-time algorithm for MFAS. ■

3.2 Relationship between Benchmarks

The first observation is that there are some instances where the best rank loss is less than any fixed distributional loss, only becoming equal to the infimum of distributional loss. For example, consider the following three-round game:

Claim 2 *There exist instances of sleeping expert allocation where* $\inf_u L_T(u)$ *is not attained and* $\inf_u L_T(u) = \min_\sigma L_T(\sigma)$.

Proof Consider the following three-round game, where each entry represents the loss of an expert at a given round, unless that expert is asleep.

	Expert 1	**Expert 2**	**Expert 3**
Round 1	asleep	1	2
Round 2	3	asleep	4
Round 3	5	4	asleep

Table 3: Simple game where the infimum of the distribution loss is not attained.

Observe that best ranking of experts is $\sigma^* = 2 - 1 - 3$, which gives rise to the best rank loss $L_T^{(rank)} = 8$. It suffices to show, given any fixed distribution $u = (u_1, u_2, u_3)$, there exists u' such that $L_T(u') \le L_T(u)$. First of all, if $u_3 > 0$, taking $u'_3 = 0$ will always decrease the loss, because every round Expert 3 is awake, its loss is greater than the other awake expert. For $u_3 = 0$, we have $u_1 = 1 - u_2$. Note that u_2 cannot be zero as otherwise the loss in Round 1 would not be well-defined (likewise with u_1 and Round 2). Thus the loss will be $1+3+5(1-u_2)+4u_2 = 9-u_2 \to 8 = L_T^{(rank)}$ asymptotically, as u_2 approaches 1. ■

To our knowledge this has not been observed in the literature. Indeed, in previous literature, the distributional benchmark has been written incorrectly as a minimum.

Corollary 1 *The definition of distribution benchmark given by (Freund et al., 1997) as* $\min_{u\in\Delta_{N-1}} L_d(u)$ *(instead of* $\inf_{u\in\Delta_{N-1}} L_d(u)$ *as defined here) is ill-defined.*

The construction in this example lends itself to the question: can we use the optimal ordering of experts to construct an optimal distribution (in the limit)? While we fall short of showing this in the general case of adversarial losses and availability, we show a similar statement when we relax one or the other to be stochastic.

3.2.1 Adversarial Losses and Availability

When the losses and experts availabilities are both adversarially chosen, we conjecture that one can construct an optimal distribution based on the optimal ordering. We define this construction below.

Definition 4 *Fix* $l_1, ..., l_T$ *loss vectors and let* $\sigma^* = \arg\min_\sigma L_T(\sigma)$. *Define the* ϵ***-rank-induced distribution*** *as:*

$$u^{(\epsilon)} = \sigma^*(1, \epsilon, ..., \epsilon^{N-1})Z_\epsilon$$

where $Z_\epsilon = \frac{1-\epsilon}{1-\epsilon^n}$ *(for normalization).*

Question 1 *Does* $\lim_{\epsilon\to 0} L_T(u^{(\epsilon)}) = \inf_u L_T(u)$ *for all* $l_1, ..., l_T \in \mathbb{R}_+^N$*?*

Together with the following lemma, this would imply that the rank benchmark is always dominated by the distribution benchmark.

Lemma 1 $\lim_{\epsilon\to 0} L_T(u^{(\epsilon)}) = \min_\sigma L_T(\sigma)$.

Proof More concretely, we would like to show:

$$\lim_{\epsilon\to 0} \sum_{t=1}^{T} \frac{1}{u^{(\epsilon)}(E_t)} \sum_{i\in E_t} u_i^{(\epsilon)} l_{t,i} = \sum_{t=1}^{T} l_{t,\sigma^*(E_t)}$$

Fix a day τ. Note that, for all $i \in E_\tau \setminus \{\sigma^*(E_\tau)\}$, we have:

$$u_i^{(\epsilon)} / u_{\sigma^*(E_\tau)}^{(\epsilon)} \leq \epsilon \to 0 \text{ as } \epsilon \to 0$$

So $u_i^{(\epsilon)}/u^{(\epsilon)}(E_\tau) \to \mathbf{1}[i = \sigma^*(E_\tau)]$. Then loss per day becomes $l_{\tau,\sigma^*(E_\tau)}$. ■

To show equality of the benchmarks, it suffices to show the following: for all loss vectors $(l_t, E_t)_{t\in[T]}$ and for all $u \in \Delta(n)$, there exists $u' \in \Delta(S_n)$ (a distribution over permutations) such that:

$$\sum_{t=1}^{T} \frac{1}{u(E_t)} \sum_{i\in E_t} l_{t,i} = \sum_{\sigma\in S_n} (u'_\sigma \sum_{t=1}^{T} l_{t,\sigma(E_t)})$$

then it is clear that the RHS is larger than $\min_\sigma L_R(\sigma)$.

We construct the requisite u' using a so-called Plackett-Luce model parameterized by u.

3.2.2 Stochastic Losses and Adversarial/Stochastic Availability

The above analysis assumed that the losses and availability of experts are both Adversarially chosen. In this section, we assume that the losses are chosen stochastically:

Claim 3 *Suppose $l_{t,i} \sim \mathbb{P}_{t,i}$ with time-invariant mean $\mu_{t,i} = \mu_i$ for all $t \in [T]$ and $i \in [N]$. Consider two separate cases: 1) $E_t \sim \mathbb{P}_{avail}$ for all $t \in [T]$, 2) E_t's are chosen Adversarially for all $t \in [T]$. In either case,*

$$\min_{\sigma} \mathbb{E}[L_T(\sigma)] = \inf_{u} \mathbb{E}[L_T(u)]$$

and the infimum is achieved as $\epsilon \to 0$ by $u^{(\epsilon)}$ the rank-induced distribution.

Proof The expected rank loss is computed for two cases as follows:

$$\mathbb{E}[L_T(\sigma)] = \sum_{t=1}^{T} \sum_{E \subset [N]} \mathbb{P}(E)\mathbb{E}[l_{t,\sigma(E)}] = T \sum_{E \subset [N]} \mathbb{P}(E)\mu_{\sigma(E)} \quad \text{(Stochastic Availability)}$$

$$\mathbb{E}[L_T(\sigma)] = \sum_{t=1}^{T} \mathbb{E}[l_{t,\sigma(E_t)}] = \sum_{t=1}^{T} \mu_{\sigma(E_t)} \quad \text{(Adversarial Availability)}$$

In either case, the expected rank loss is minimized by $\sigma^* = \text{sort}_i\{\mu_i\}$ in descending order. That is because always choosing the smallest-average-loss expert among all available experts in any set E (or E_t) minimizes the total loss in expectation. Now consider the expected distribution loss for some $u \in \Delta_{N-1}$.

$$\mathbb{E}[L_T(u)] = \sum_{t=1}^{T} \sum_{E \subset [N]} \mathbb{P}(E) \sum_{i \in E} \frac{u_i}{u(E)} \mathbb{E}[l_{t,i}] = T \sum_{E \subset [N]} \mathbb{P}(E) \sum_{i \in E} \frac{u_i}{u(E)} \mu_i \quad \text{(Stochastic Availability)}$$

$$\mathbb{E}[L_T(u)] = \sum_{t=1}^{T} \sum_{i \in E_t} \frac{u_i}{u(E_t)} \mathbb{E}[l_{t,i}] = \sum_{t=1}^{T} \sum_{i \in E_t} \frac{u_i}{u(E_t)} \mu_i \quad \text{(Adversarial Availability)}$$

In either case, to minimize the expected cumulative loss, one should choose the distribution u such that $\frac{u_i}{u(E)}\mu_i$ (and $\frac{u_i}{u(E_t)}\mu_i$) is minimized for each set E (and E_t). The more the expert with the smallest-average-loss in each given set E (and E_t) is weighted, the smaller $\frac{u_i}{u(E)}\mu_i$ (and $\frac{u_i}{u(E_t)}\mu_i$) will get. To accomplish that, the weight vector u should be chosen such that the normalized weight of the smallest-average-loss expert in any set E (and E_t) dominates all other normalized weights of experts in that E (and E_t). This is best accomplished by choosing the rank-induced distribution $u^{(\epsilon)} = \sigma^*(1, \epsilon, ..., \epsilon^{N-1})Z_\epsilon$ (where the indices of $u^{(\epsilon)}$ follows the best ranking σ^*) for small ϵ.

As $\epsilon \to 0$, $u_i^{(\epsilon)}/u^{(\epsilon)}(E_t) \to \mathbf{1}(i = \sigma(E_t))$, and the distributional loss will approach the rank loss. Thus, $\min_\sigma \mathbb{E}[L_{T,\sigma}] = \inf_u \mathbb{E}[L_{T,u}]$. ■

3.2.3 Adversarial Losses and Stochastic Availability

In this section, we assume that the losses are adversarially chosen but the expert availabilities are chosen stochastically:

Claim 4 *Suppose $l_{t,i}$'s for all $t \in [T]$ and $i \in [N]$ are chosen adversarially and $E_t \sim \mathbb{P}_{avail}$ for all $t \in [T]$. Then,* $\min_\sigma \mathbb{E}[L_T(\sigma)] = \inf_u \mathbb{E}[L_T(u)]$.

Proof The rank loss is given by:

$$
\begin{aligned}
L_T(\sigma) &= \sum_{t=1}^{T} \sum_{E \subset [N]} \mathbb{P}(E) l_{t,\sigma(E)} \\
&= \sum_{E \subset [N]} \mathbb{P}(E) \sum_{t=1}^{T} l_{t,\sigma(E)} \\
&= \sum_{E \subset [N]} \mathbb{P}(E) L_{T,\sigma(E)} \quad \text{where } L_{T,\sigma(E)} = \sum_{t=1}^{T} l_{t,\sigma(E)}
\end{aligned}
$$

Note that $\mathbb{P}_{\text{avail}}$ is fixed. Then to minimize the rank loss, among the experts in any set E, one must always follow the expert whose total loss over all rounds is the smallest. To achieve that, one must rank the experts in the increasing order of their total losses over all rounds. That is, $\sigma^* = \text{sort}_i(\sum_{t=1}^{T} L_{t,i})$ in increasing order. See Claim 5 for a more detailed proof of this idea.

Meanwhile, the distributional loss is:

$$
\begin{aligned}
L_T(u) &= \sum_{t=1}^{T} \sum_{E \subset [N]} \mathbb{P}(E) \frac{1}{u(E)} \sum_{i \in E} u_i l_{t,i} \\
&= \sum_{E \subset [N]} \sum_{i \in E} \mathbb{P}(E) \frac{u_i}{u(E)} \sum_{t=1}^{T} l_{t,i} \\
&= \sum_{E \subset [N]} \mathbb{P}(E) \sum_{i \in E} \frac{u_i}{u(E)} L_{T,i} \quad \text{where } L_{T,i} = \sum_{t=1}^{T} l_{t,i}
\end{aligned}
$$

To minimize the distributional loss, for any set E, one must put as much weight as possible to the expert in the set E with the smallest total loss over all rounds, $L_{T,i}$. Thus, again it suffices to choose the rank-induced distribution $u^{(\epsilon)} = \sigma^*(1, \epsilon, ..., \epsilon^{N-1}) Z_\epsilon$, hence $\min_\sigma \mathbb{E}[L_{T,\sigma}] = \inf_u \mathbb{E}[L_{T,u}]$.

■

3.2.4 Adversarial Losses and Availability

Before we tackle the general adverserial case, it is instructive to consider the case of the uniform distribution over experts.

Claim 5 *Let u be the uniform distribution over $[n]$. Then:*

$$L_D(u) > \min_\sigma L_R(\sigma)$$

Proof Let $u' = \mathrm{Unif}(S_n)$ and $u = \mathrm{Unif}([n])$. We have:

$$\begin{aligned} L_D(u) &= \sum_t \sum_{i \in E_t} l_{t,i}/|E_t| && (2) \\ &= \sum_t \mathbb{E}_{\sigma \sim u'} l_{t,\sigma(E_t)} && (3) \\ &= \mathbb{E}_{\sigma \sim u'} \sum_t l_{t,\sigma(E_t)} && (4) \\ &= \mathbb{E}_{\sigma \sim u'} L_R(\sigma) \geq \min_\sigma L_R(\sigma) && (5) \end{aligned}$$

Step (2) follows from the fact that $\sigma(E_t) = \operatorname{argmin}_{i \in E_t} \sigma(i) = \sigma^{-1}(\min_{i \in E_t} \sigma(i))$, i.e. it outputs the top-ranked element in E_t. A uniform distribution over σ makes $\sigma(E_t)$ uniform over E_t. ■

The goal is to generalize this argument. Note that we make the transformation, without loss of generality, from arbitrary loss vectors and expert availabilities, to losses collated based on expert availability: $(l_t, E_t)_{t \leq T} \mapsto (\tilde{l}_E)_{E \subset [n]}$, where $\tilde{l}_E = \sum_{t: E_t = E} l_t$. The generalized argument would look something like the following:

$$\begin{aligned} L_D(u) &= \sum_{E \subset [n]} \frac{1}{u(E)} \sum_{i \in E} u_i \tilde{l}_{E,i} && (6) \\ &= \sum_{E \subset [n]} \mathbb{E}_{i \sim u}[\tilde{l}_{E,i} \mid i \in E] && (7) \\ &:= \sum_{E \subset [n]} \mathbb{E}_{i \sim u_E}[\tilde{l}_{E,i}] && (8) \\ (\text{WTS } \exists u' \text{ over } S_n) \quad &= \sum_{E \subset [n]} \mathbb{E}_{\sigma \sim u'}[\tilde{l}_{E,\sigma(E)}] && (9) \\ &= \mathbb{E}_{\sigma \sim u'} \sum_{E \subset [n]} \tilde{l}_{E,\sigma(E)} && (10) \\ &= \mathbb{E}_{\sigma \sim u'} L_R(\sigma) && (11) \end{aligned}$$

Note that a distribution over $\sigma \in S_n$ induces 2^n distributions, one for each subset E, via $\sigma \mapsto \sigma(E)$. We are basically asking about the surjectivity of this map. More precisely: given $u \in \Delta_{[n]}$ and all of its marginals $u_E \in \Delta_E$, can we construct $u' \in \Delta_{S_n}$ such that $\sigma(E) = \sigma^{-1} \min_{i \in E} \sigma(i)$ has distribution u_E? It turns out, the answer is yes.

Claim 6 *For all $\{l_E\}_{E \subset [n], t \leq T}$ and $u \in \Delta(n)$, there exists $u' \in \Delta(S_n)$ such that:*

$$\mathbb{E}_{i \sim u}[l_{E,i} \mid i \in E] = \mathbb{E}_{\sigma \sim u'}[\tilde{l}_{E,\sigma(E)}]$$

This shows equivalence of rank and distribution losses.

Proof $u' \in \Delta(S_n)$ is given by a Plackett-Luce model with weights given by u. Namely:

$$\mathbb{P}(\sigma) = \prod_{i=1}^{n} \frac{u_{\sigma_i}}{\sum_{j \in [n] \setminus S_i} u_{\sigma_j}}$$

where $S_i = \{1, ..., j-1\}$. Once can check that this is indeed a distribution and that it satisfies the following nice property:

$$\mathbb{P}(\sigma(E) = i) = \mathbb{P}(\text{“pick } i \text{ from } E\text{”}) = \frac{u_i}{\sum_{j \in E} u_j}$$

Therefore:

$$\begin{aligned}
\mathbb{E}_{\sigma \sim u'}[\tilde{l}_{E,\sigma(E)}] = \sum_{\sigma \in S_n} u'_\sigma \tilde{l}_{E,\sigma(E)} &= \sum_{i \in [n]} \tilde{l}_{E,\sigma(E)} \sum_{\sigma : \sigma(E) = i} u'_\sigma \\
&= \sum_{i \in [n]} \tilde{l}_{E,\sigma(E)} \mathbb{P}(\sigma(E) = i) \\
&= \sum_{i \in [n]} \tilde{l}_{E,\sigma(E)} \frac{u_i}{\sum_{j \in E} u_j} \\
&= \mathbb{E}_{i \in u_E}[\tilde{l}_{E,i}]
\end{aligned}$$

■

We can show the margin-matching property of the Plackett-Luce model explicitly for three experts. Let $E = \{1, 2\}$. Then:

$$\begin{aligned}
\sum_{\sigma : \sigma(\{1,2\}) = 1} u'_\sigma &= u'_{123} + u'_{132} + u'_{312} \\
&= \frac{u_1}{u_1 + u_2 + u_3} \frac{u_2}{u_2 + u_3} + \frac{u_1}{u_1 + u_2 + u_3} \frac{u_3}{u_2 + u_3} + \frac{u_3}{u_1 + u_2 + u_3} \frac{u_1}{u_1 + u_2} \\
&= \frac{u_1}{u_1 + u_2 + u_3} \left(\frac{u_2}{u_2 + u_3} + \frac{u_3}{u_2 + u_3} + \frac{u_3}{u_1 + u_2} \right) \\
&= \frac{u_1}{u_1 + u_2 + u_3} \left(1 + \frac{u_3}{u_1 + u_2} \right) \\
&= \frac{u_1}{u_1 + u_2 + u_3} \left(\frac{u_1 + u_2 + u_3}{u_1 + u_2} \right) \\
&= \frac{u_1}{u_1 + u_2}
\end{aligned}$$

4 Second-Order Bounds for Sleeping Experts

In this section, we seek to take the usual bounds for sleeping experts and replace the dependence on T with a measure of variation in the loss vectors:

$$\mathrm{VAR}_T^{\max} = \max_{t \le T} \{\mathrm{VAR}_{t,i_t^*}\} = \max_{t \le T} \left\{ \sum_{s=1}^{t} (l_{s,i_t^*} - \mu_{t,i_t^*})^2 \right\} \tag{12}$$

where $i_t^* = \text{argmin}_i\{\sum_{s=1}^t l_{t,i}\}$ is the best expert up until round t and $\mu_{t,i} = \frac{1}{t}\sum_{s=1}^t l_{s,i}$. In the usual (non-sleeping) online allocation problem, one can achieve $O(\sqrt{\text{VAR}_T^{\max}\log N})$ regret, using a variant of Hedge developed by (Hazan and Kale, 2010).

For the sake of completeness, we review the Variation MW algorithm and give a brief sketch of how its regret bound arises.

Algorithm 1 Variation MW

Initialize $w_{1,i} = 1$ for all $i \in [N]$.
for day $t = 1, ..., T$ **do**
  Play $p_t = w_t / \|w_t\|_1$.
  Update weights:

$$w_{t+1,i} = w_{t,i} \exp\Big(-\eta \underbrace{(l_{t,i} + 4\eta(l_{t,i} - \mu_{t,i})^2)}_{\tilde{l}_{t,i}}\Big)$$

  Incur loss $\langle p_t, l_t\rangle$.
end for

Theorem 2 *Let $\eta = \min\{1/10, \sqrt{\log N/\text{VAR}_T^{\max}}\}$. Then Variation MW achieves.*

$$\sum_{t=1}^T \langle p_t, l_t\rangle - \min_i \sum_{t=1}^T l_{t,i} \le 8\sqrt{\text{VAR}_T^{max}\log N} + 10\log N$$

Since $\text{VAR}_T^{\max}$ increases with T, the correct η can be learned via the doubling trick.

The proof proceeds as follows:

- Instead of analyzing $\text{VAR}_{t,i}$ directly, analyze

$$Q_{t,i} = \sum_{s=1}^{t-1}(l_{s,i} - \mu_{s,i})^2$$

  which is an asymptotic proxy for $\text{VAR}_{t,i}$. Specifically, $Q_t^{\max} \le 4\,\text{VAR}_t^{\max}$.

- Define $g_t = \tilde{l}_t - \alpha_t \mathbf{1}$ where $\alpha_t = \mu_{t,i_t^*} + 4\eta Q_{t,i_t^*}/t$. Apply the standard Hedge guarantee to g_t as the sequence of loss vectors. Since $\exp(-\eta\sum_t g_{t,i})$ and $\exp(-\eta\sum_t \tilde{l}_{t,i})$ only differ by a constant factor (independent of i), the weight updates are equivalent, so the guarantee gives us:

$$\sum_{t=1}^T \langle \tilde{l}_t, p_t\rangle - \sum_{t=1}^T \tilde{l}_{t,i_T^*} \le \eta\sum_{t=1}^T \langle g_t^2, p_t\rangle + \frac{\log N}{\eta}$$

  Simplifying gives us an upper bound of:

$$\eta\sum_{t=1}^T \Big\langle g_t^2 - 4(f_t - \mu_t)^2, p_t\Big\rangle + 4\eta(Q_T^{\max} + 1) + \frac{\log N}{\eta}$$

- From here, it suffices to show the sum in the first term is on the order of $Q_T^{\max}$. We refer the reader to (Hazan and Kale, 2010) for the details of this calculation.

4.1 Stochastic Availability, Adversarial Loss

It turns out there is an efficient algorithm with $O(\sqrt{T \log N})$ regret for sleeping experts in the setting where the adversary is oblivious, the availability of experts is chosen according to a fixed probability distribution, and the losses are chosen Adversarially. This is shown in (Kanade et al., 2009). We get the same guarantee for sleeping experts with stochastic availability. Our proof is similar to that of (Kanade et al., 2009) but instead of applying the usual Hedge guarantee, apply the variation-adaptive guarantee from (Hazan and Kale, 2010). We call our algorithm Variation MW for Sleeping Setting (VMWS).

Algorithm 2 Variation MW for Sleeping Setting (VMWS)

Initialize $w_{1,i} = 1$ for all $i \in [N]$.
for day $t = 1, ..., T$ **do**
  Observe $E_t \sim \mu_{avail}$
  Play q_t where $q_{t,i} = w_{t,i} / \sum_{i \in E_t} w_{t,a}$ for $i \in E_t$.
  Play i_t, incur l_{t,i_t} loss.
  Observe full loss vector l_t.
  Update weights:

$$w_{t+1,i} = \begin{cases} w_{t,i} \exp\Big(- \eta(l_{t,i} + 4\eta(l_{t,i} - \mu_{t,i})^2)\Big) & i \in E_t \\ w_{t,i} & i \notin E_t \end{cases}$$

  where $\mu_t = (1/t) \sum_{s=0}^{t-1} l_t$.
end for

Theorem 3 *If* $\eta = \min\{1/10, \sqrt{\log N / \mathrm{VAR}_T^{\max}}\}$, *algorithm VMWS has regret*

$$\mathbb{E}\Big[\sum_{t=1}^{T} l_{t,i_t} - \min_{\sigma \in S_N} \mathbb{E}_{E_t} \sum_{t=1}^{T} l_{t,\sigma(E_t)}\Big] \leq O\Big(\sqrt{\mathrm{VAR}_T^{max} \log N}\Big)$$

where the other expectation is taken with respect to the inherent randomness in the algorithm.

Note, by Jensen's inequality, that

$$\min_{\sigma \in S_N} \mathbb{E}_{E_t} \sum_{t=1}^{T} l_{t,\sigma(E_t)} \geq \mathbb{E}_{E_t} \min_{\sigma \in S_N} \sum_{t=1}^{T} l_{t,\sigma(E_t)} = \mathbb{E}[L_T^{(rank)}]$$

So a regret bound with respect to $\mathbb{E}[L_T^{(\mathrm{rank})}]$ is a bit stronger than Theorem 8. The reason we use this slightly modified notion of rank regret is that it is easier to analyze. Consider the following claim (not directly related to the algorithm, just a statement about the nature of stochastic availability). It tells us that, under a fixed distribution of expert availability, the best post-hoc ordering of experts is given by simply sorting the post-hoc losses.

Claim 7 *Let* $\sigma^* = \mathrm{argmin}_\sigma \mathbb{E}_{E_1,...,E_t} \sum_{t=1}^{T} l_{t,\sigma(E_t)}$. *Then* $\sigma^* = \mathrm{sort}_i(\sum_{t=1}^{T} l_{t,i})$ *(ascending)*

Proof Let $p_{\sigma,i} = \mathbb{P}_E(\sigma(E) = i) = \mathbb{P}_E(\min_{j\in E}\sigma(j) = i)$, where $E \sim \mu_{avail}$. Note that $p_{\text{id},i} = \mathbb{P}_E(\min E = i)$, so $p_{\text{id},1} \geq ... \geq p_{\text{id},n}$. This implies that p_σ is ordered in descending order by σ.

$$\begin{aligned}
\mathbb{E}_{E_t}\sum_{t=1}^T l_{t,\sigma(E_t)} &= \sum_{t=1}^T \sum_{E\subset[N]} \mathbb{P}(E_t = E)\cdot l_{t,\sigma(E)} \\
&= \sum_{t=1}^T \sum_{i\in[N]} \mathbb{P}(\sigma(E_t) = i)\cdot l_{t,i} \\
&= \sum_{t=1}^T \langle p_\sigma, l_t\rangle = \langle p_\sigma, \sum_{t=1}^T l_t\rangle
\end{aligned}$$

Without loss of generality, suppose $\sum_{t=1}^T l_t$ is arranged in ascending order. Note that σ^* minimizing the above expression is such that p_{σ^*} matches its largest values with the smallest values of $\sum_t l_t$ (since everything is non-negative). Hence, $\sigma^* = \text{sort}_i(\sum_{t=1}^T l_{t,i})$ (p_σ descending, loss ascending). ■

Proof Consider a fixed action set $E \subset [N]$. Let $i^* = \text{argmin}_{i\in E}\sum_{t=1}^T l_{t,i}$. Applying the bound from Theorem 4 of (Hazan and Kale, 2010) for the Variation MW algorithm, we have the following regret bound:

$$\sum_{t=1}^T \sum_{i\in A} q_{t,i} l_{t,i} - \sum_{t=1}^T l_{t,i^*} \leq \underbrace{8\sqrt{\text{VAR}_T^{max}\log N} + 10\log N}_{\textsf{bound}}$$

Let σ^* be the post-hoc best action list, given by $\sigma^* = \text{sort}_i(\sum_{t=1} l_{t,i})$ as per Claim 1. Then $i^* = \sigma^*(A)$ where $\sigma^* = \text{argmin}_\sigma \mathbb{E}_{E_t}[\sum_{t=1}^T l_{t,\sigma(E_t)}]$

$$\begin{aligned}
\mathbb{E}_{i_t}\Big[\sum_{t=1}^T l_{t,i_t} - \min_{\sigma\in S_N}\mathbb{E}_{E_t}\Big[\sum_{t=1}^T l_{t,\sigma(E_t)}\Big]\Big] &= \mathbb{E}_{i_1,\dots,i_T}\sum_{t=1}^T \mathbb{E}_{E_t}\Big[\sum_{i\in E_t}\mathbb{E}_{i_t|i_1,\dots,i_{t-1}}\Big[l_{t,i_t}\Big] - l_{t,\sigma^*(E_t)}\Big] \\
(\text{Claim 1}) &= \sum_{t=1}^T \mathbb{E}_{E_t}\Big[\sum_{i\in E_t} q_{t,i} l_{t,i} - l_{t,i^*}\Big] \\
&= \sum_{t=1}^T \sum_{E\subset[N]} \mathbb{P}(E_t = E)\Big[\sum_{i\in E_t} q_{t,i} l_{t,i} - l_{t,i^*}\Big] \\
&= \sum_{E\subset[N]} \mathbb{P}(E_t = E)\Big(\sum_{t=1}^T \sum_{i\in E_t} q_{t,i} l_{t,i} - l_{t,i^*}\Big) \\
&\leq \sum_{E\subset[N]} \mathbb{P}(E_t = E)\cdot \textsf{bound} \\
&= \textsf{bound}
\end{aligned}$$

Note that in first steps, we use tower property to be precise about the nature of the randomness in the algorithm. ■

4.2 Adversarial Availability and Loss

There is a simple but inefficient algorithm to deal with the case where availability of experts is adversarial. The idea is to transform the problem into an insomniac setting with $N!$ experts, each one corresponding to a permutation on N elements. Each expert predicts $\sigma(E_t)$ for each round t.

Theorem 4 *Using Variation MW with $N!$ weights, one per permutation on $[N]$, we achieve:*

$$\mathbb{E}\Big[\sum_{t=1}^{T} l_{t,i_t} - \min_{\sigma} \sum_{t=1}^{T} l_{t,\sigma(E_t)}\Big] \leq O\Big(\sqrt{\mathrm{VAR}_T^{max} N \log N} + N \log N\Big)$$

where i_t is the action chosen by the algorithm at time t and the expectation is made over the randomness in the algorithm.

Proof An immediate application of the Variation MW algorithm achieves:

$$\sum_{t=1}^{T} \sum_{\sigma \in S_N} p_\sigma l_{t,\sigma(E_t)} - \min_{\sigma} \sum_{t=1}^{T} l_{t,\sigma(E_t)} \leq 8\sqrt{\mathrm{VAR}_T^{max} \log N!} + 10 \log N!$$

Observe that $\log(N!) \leq N \log N$, and the first term is the expectation. ■

5 Conclusion

There are a few interesting open questions that follow naturally from this work. First of all, can we extend the beyond worst-case analysis of sleeping experts? While the application of second-order bounds is fairly immediate, what are the challenges of applying quantile bounds? Could we extend the work of (Chaudhuri et al., 2009) to develop parameter-free hedging for sleeping experts?

There is also more work to be done in regard to the comparison between rank and distribution loss for sleeping experts. It is left as conjecture whether these losses coincide for the case when both losses and action availability are adversarial. It would also be interesting to settle whether computing the rank loss is truly an NP-hard problem.

Acknowledgments

The authors would like to thank Daniel Hsu for his feedback in the writing process, and for providing the idea for the reduction from minimum feedback arc set.

References

Avrim Blum and Yishay Mansour. From external to internal regret. *Journal of Machine Learning Research*, 8(6), 2007.

Kamalika Chaudhuri, Yoav Freund, and Daniel J Hsu. A parameter-free hedging algorithm. *Advances in neural information processing systems*, 22, 2009.

Yoav Freund, Robert E Schapire, Yoram Singer, and Manfred K Warmuth. Using and combining predictors that specialize. In *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*, pages 334–343, 1997.

Pierre Gaillard, Gilles Stoltz, and Tim Van Erven. A second-order bound with excess losses. In *Conference on Learning Theory*, pages 176–196. PMLR, 2014.

Elad Hazan and Satyen Kale. Extracting certainty from uncertainty: Regret bounded by variation in costs. *Machine learning*, 80:165–188, 2010.

Varun Kanade, H Brendan McMahan, and Brent Bryan. Sleeping experts and bandits with stochastic action availability and adversarial rewards. In *Artificial Intelligence and Statistics*, pages 272–279. PMLR, 2009.

Robert Kleinberg, Alexandru Niculescu-Mizil, and Yogeshwer Sharma. Regret bounds for sleeping experts and bandits. *Machine learning*, 80(2):245–272, 2010.

Wouter M Koolen and Tim Van Erven. Second-order quantile methods for experts and combinatorial games. In *Conference on Learning Theory*, pages 1155–1175. PMLR, 2015.

Nick Littlestone and Manfred K Warmuth. The weighted majority algorithm. *Information and computation*, 108(2):212–261, 1994.