Online Learning with Sleeping Experts and Feedback Graphs

Corinna Cortes 1, Giulia DeSalvo 1, Claudio Gentile 1, Mehryar Mohri 1,2, Scott Yang 3

Abstract

We consider the scenario of online learning with sleeping experts, where not all experts are available at each round, and analyze the general framework of learning with feedback graphs, where the loss observations associated with each expert are characterized by a graph. A critical assumption in this framework is that the loss observations and the set of sleeping experts at each round are independent. We first extend the classical sleeping experts algorithm of Kleinberg et al. (2008) to the feedback graphs scenario, and prove matching upper and lower bounds for the sleeping regret of the resulting algorithm under the independence assumption. Our main contribution is then to relax this assumption, present a more general notion of sleeping regret, and derive a general algorithm with strong theoretical guarantees. We apply this new framework to the important scenario of online learning with abstention, where a learner can abstain from making a prediction at the price of a certain cost. We empirically validate our algorithm against multiple online abstention algorithms on several real-world datasets, showing substantial performance improvements.

1. Introduction

Sequential decision making under uncertainty is an important and widely studied area of machine learning. In the standard online learning framework (Cesa-Bianchi & Lugosi, 2006), at each round, the learner selects an action out of a finite set and incurs some loss associated with that action. The learner’s goal is to minimize her regret over a finite number of rounds, that is the difference between her cumulative loss and that of the best static action in hindsight.

Online learning with feedback graphs is a general framework for online learning where the action losses that are observable to the learner are modelled by graphs. This framework was first introduced by Mannor & Shamir (2011) and later analyzed by several other authors (Caron et al. (2012); Buccapatnam et al. (2014); Wu et al. (2015); Alon et al. (2013); Kocák et al. (2014); Alon et al. (2015); Cohen et al. (2016); Kocák et al. (2016); Tossou et al. (2017); Liu et al. (2018); Yun et al. (2018)). Given a directed feedback graph, an edge from $i$ to $j$ indicates that the loss of $j$ is observed if expert $i$ is selected by the algorithm. Such partial observability setups cover a variety of applications (e.g. in web advertising, a user who clicks on an ad reveals information about related ads). The classical settings of full information (Littlestone & Warmuth, 1994) and bandit learning (Auer et al., 2002) online learning are special instances corresponding to a fully connected graph and a graph admitting only self-loops, respectively. In general, these graphs can be either fixed or time-varying, and they can also even be stochastic.

Distinct from the feedback graph framework, online learning has also been studied in a setting in which the actions available to the learner can change at different rounds. This scenario is called the sleeping experts setting. It was analyzed first by (Kleinberg et al., 2008) and subsequently by (Kanade et al., 2009; Kanade & Steinke, 2014). In this framework, at each round the environment determines a set of available actions either stochastically or adversarially. This model of online learning can arise, e.g., in routing network problems, where some routes may be unavailable due to either random router crashes (stochastic case) or an illicit agent (adversarial case).

Restricted feedback and restricted action sets are two closely related ideas, and many applications can actually be formalized as a combination of both sleeping experts and feedback graphs. For instance, consider again the scenario of web advertising described above, where a learner has to decide which ads to display. Some ads may not be available at each round, implying that the experts are sleeping, and at the same time, related ads may have similar rewards, so that the feedback between some of the ads should be shared. Similarly, in e-commerce from the seller’s perspective, some items may be out of stock (sleeping experts), and the rewards for one item may be similar to the rewards of others (feedback graph). As a third example, sensor networks are
We present guarantees for our new notion of regret that is against multiple online abstention algorithms on several real-world datasets, showing that substantial improvements are also achieved empirically.

2. Preliminaries

We denote by $\mathcal{X}$ the input space, by $\mathcal{Y}$ the output space, and by $\mathcal{D}$ a probability distribution over $\mathcal{X} \times \mathcal{Y}$. Let $\mathcal{E}$ denote the family of experts (or actions): $\mathcal{E} = \{\xi_j : j \in [K]\}$, where $[K] = \{1, \ldots, K\}$, and let $L : \mathcal{E} \times (\mathcal{X} \times \mathcal{Y}) \to [0, 1]$ be a loss function.

We consider the scenario of online learning with side-information modeled by feedback graphs introduced by Mannor & Shamir (2011). For any $t \in [T]$, a feedback graph $G^t = (V^t, E^t)$ is a directed graph over the set of experts with indices $i \in [K]$, which admits an edge from $i$ to $j$ if the loss of $j$ is observed by the algorithm when it selects expert $i$ at round $t$. Let $N^t_i$ denote the out-neighborhood of $i$ at time $t$, that is the set of vertices $j \in [K]$ for which $G^t$ admits an edge from $i$ to $j$. We will specify our assumptions behind how the graphs $G^t$ are generated in future sections.

We consider a stochastic setting of online learning with feedback graphs, which admits the following learning protocol. At each round $t \in [T]$, a pair $(x_t, y_t) = (z_t) \in \mathcal{X} \times \mathcal{Y}$ is drawn i.i.d. from $\mathcal{D}$. The learner receives the input $x_t \in \mathcal{X}$ drawn i.i.d. according to the marginal distribution associated with $\mathcal{D}$, selects an index $I_t \in [K]$ corresponding to an expert $\xi_{I_t} \in \mathcal{E}$, incurs the loss $L(\xi_{I_t}, z_t)$, and observes the loss of every expert in the out-neighborhood of $I_t$, that is, $L(\xi_j, z_t)$, with $j \in N^t_{I_t}$. Note that the full information setting corresponds to the case where, for all $t$, $G^t$ is the fully-connected graph, while the multi-armed bandit model matches the case where $G^t$ only contains self-loops. In what follows, we assume that the loss of the expert selected is always observed. Thus, for all $t \in [T]$, $G^t$ contains self-loops at all nodes: $i \in N^t_i$, for all $i \in [K]$.

We also adopt the sleeping experts framework introduced by Kleinberg et al. (2008). In this setting, at each round $t$, the environment also generates a set $A^t \subseteq [K]$ of available (or awake) experts. We denote by $\bar{A}$ the set of all awake sets that can be possibly generated, which is a subset of the power set of $[K]$.

Note that, while the sleeping expert and feedback graph frameworks are related, one does not subsume the other. Awake sets determine which experts can be chosen by the learner, while feedback graphs determine which losses can be observed by the learner. Thus, there is no distribution over awake sets that can mimic a feedback graph scenario, and there is no set of feedback graphs that can lead to a sleeping experts setting.

Since not all experts or actions are available at each round...
in the sleeping experts setting, the best expert in hindsight, used in the standard notion of regret, is not a realistic benchmark in this setting. Instead, the notion of sleeping regret was introduced by Kleinberg et al. (2008). This notion of regret considers the difference between the cumulative loss of the algorithm and that of the best ordering of the experts, where, at each round, the expert with the most favorable rank among those awake is selected.

In our setting, the sleeping regret of an algorithm ALG can be defined as follows:

\[ \sum_{t=1}^{T} \mathbb{E}[L(\xi_t, z_t)] - \sum_{t=1}^{T} \mathbb{E}[L(\xi_{\sigma(A_t)}, z_t)], \]

where \( I_t \) is the index of the expert selected by ALG at round \( t \), and \( \sigma(A_t) \) the index of the expert with the smallest expected loss among those in \( A_t \). Here, the expectations are taken over the algorithm’s actions \( I_t \) (for a randomized algorithm), over the choice of \( z_t \sim \mathcal{D} \), and over the generation of the awake sets \( A_t \), when they are generated stochastically.

We restrict our study to the case where the awake sets are generated stochastically, possibly based on \( z_t \), and focus on two separate scenarios, each requiring a different approach:

1. one where the awake sets are statistically independent of the losses, such that \( A_t \) is independent from \( z_t \) (Section 3);
2. one where the awake sets and the losses are statistically dependent (Section 4).

The notion of sleeping regret previously described was introduced by Kleinberg et al. (2008) for the first scenario, where the awake sets are independent of the losses. As we shall see, in the dependent case, this notion is no longer pertinent. Thus, we will generalize that expression and define a new notion of sleeping regret suitable for the dependent case.

3. Independent losses and awake sets

The independence between awake sets and losses is a crucial assumption in the study of online learning with sleeping experts by Kleinberg et al. (2008). Under this assumption, the authors presented an algorithm, called AUER, with tight theoretical guarantees. The algorithm is based on the classical Upper Confidence Bound (UCB) approach (Auer et al., 2002). It maintains a set of lower confidence bounds on the expected loss of each expert and, at each round, chooses the expert with the lowest confidence bound from the set of available experts. AUER is designed for the bandit setting, that is, when only the loss of the chosen expert is observed.

In this section, we present an extension of AUER to the feedback graph scenario, while assuming that the losses and awake sets are statistically independent. In this section, we also assume that the feedback graph \( G^t \) depends only on information up to time \( t-1 \); in particular, \( G^t \) does not depend on the losses \( L(\xi_j, z_t) \) generated at time \( t \). To be consistent with the notion of sleeping experts, we further assume that the graph \( G^t \) contains only vertices in \( A_t \), although this does not affect the proofs. The pseudocode of our algorithm, AUER-N, which stands for AUER with Neighbors, is given in Algorithm 1.

The idea behind the design of AUER-N is to update the time-\( t \) estimate \( \hat{\mu}_j(t) \) of the expected loss of all experts with index \( j \) in the out-neighborhood \( N_t^j \) of the chosen expert \( I_t \) at every round. These out-neighborhoods are determined by the time-\( t \) feedback graph \( G^t \). As with AUER, the algorithm selects the awake expert with the smallest confidence bound.

We denote by \( \mu_j = \mathbb{E}[L(\xi_j, z)] \) the expected loss of expert \( \xi_j \), and assume an indexing consistent with the ranking of these losses: \( \mu_1 < \mu_2 < \cdots < \mu_K \). For any \( i < j \), we denote by \( \Delta_{i,j} = \mu_j - \mu_i \) the decrease in expected loss from \( \xi_j \) to \( \xi_i \), and use the convention \( \Delta_{i,i} = 0 \) for any \( j \in [K] \). We also denote by \( T_j(t) \) the number of times expert \( \xi_j \) is selected by the algorithm up to time \( t \), and by \( Q_j(t) \) the number of times the loss of expert with index \( j \) is observed. The theorem below gives a bound on the sleeping regret of AUER-N in terms of \( T_j(t) \) and \( Q_j(t) \). These quantities are both algorithm-dependent, but, by definition, the ratio \( T_j(t)/Q_j(t) \) is bounded by one and can be far smaller for dense graphs.

**Theorem 1** Assume that, for any \( t \in [T] \), the feedback graph \( G^t \) depends only on information up to time \( t - 1 \), and that the awake sets \( A_t \) are generated i.i.d., independently of the loss values \( L(\xi_j, z_t) \), \( j \in [K] \). Then, the sleeping regret of AUER-N after \( T \) rounds is upper bounded as follows:

\[ \sum_{j=1}^{K} \Delta_{j-1,j} \leq 4 \log T \leq \max_{t \in [T]} Q_j(t) \] + \[ \sum_{j=1}^{K} \Delta_{j-1,j}. \]

The proof of this theorem and that of all other results are
Online Learning with Sleeping Experts and Feedback Graphs

given in the appendix. Since \( \max_t \frac{T_j(t)}{Q_j(t)} \) is upper-bounded by one for all \( j \in [K] \), the sleeping regret bound of Auer-N is always more favorable than that of Auer. In particular, if the number of times a learner choses an expert is equal to the number of times that expert was observed, that is, \( T_j(t) = Q_j(t) \), for all \( j \) (as in the standard bandit setting), then we recover the sleeping regret bound of Auer (Kleinberg et al., 2008). On the other hand, in the full information setting, when \( A_t = r \) we have \( Q_j(t) = t \) for all \( j \) and \( t \), and \( \sum_{j \in [K]} T_j(t) = t \). Thus, when the gap terms \( \Delta_{j,1} \) are all comparable, the algorithm achieves an improvement in the regret bound by a factor of \( \frac{1}{K} \), which, naturally, is a consequence of the \( K \) times more feedback received at each round, compared to the bandit setting.

We complement Theorem 1 by proving lower bounds showing that the regret of Auer-N is information-theoretically optimal, at least in the bandit scenario where the feedback graphs \( G^i \) only contain self-loops. In particular, we extend the lower bound of Kleinberg et al. (2008), which holds in the case of adversarially chosen awake sets, to the case where they are generated stochastically.

**Theorem 2** Under the assumptions of Theorem 1, assume that the graphs \( G^i \) only contain self-loops (bandit setting), and let \( \phi \) be an online algorithm for the multi-armed bandit problem that never picks a suboptimal expert more than \( o(T^\alpha) \) times over the course of \( T \) rounds for every \( \alpha > 0 \). Then, there exists a distribution according to which awake sets \( A^i \) are drawn i.i.d. and for which the sleeping regret of algorithm \( \phi \) is at least \( \Omega \left( \sum_{j=2}^{K} \frac{1}{\Delta_{j,1}} \log(T) \right) \) for large enough \( T \) depending on \( (\mu_i)_i \in [K] \).

4. Dependent losses and awake sets

In this section, we relax the assumption that the awake sets and losses are independent. We introduce a more general notion of sleeping regret, called generalized sleeping regret and argue that it is a more relevant notion of regret in the dependent setting than the standard sleeping regret defined in Kleinberg et al. (2008). We then present a sleeping experts algorithm based on the UCB algorithm of Auer et al. (2002), which we call UCB-SLG (UCB with SLG Learning Graphs), that exploits feedback graphs and admits a favorable bound for the generalized sleeping regret. This will in turn pave the way for the application to the scenario of online abstention covered in Section 5, where the losses and awake sets are dependent and, where a natural notion of feedback graph over experts can be exploited.

4.1. Generalized sleeping regret

To see why the dependence between losses and awake sets invalidates the classical notion of sleeping regret, recall that the standard definition of sleeping regret (see Section 2) uses \( \mathbb{E}[L(\xi, z)] \) to compare the expected loss of the chosen expert against the awake expert with the smallest expected loss. Then, consider the natural scenario where the input space \( X \) is the real line, and where an expert is awake only when \( x > 0 \), thereby making losses and awake sets both depend on \( x \). Notice that the loss of this expert can only be (potentially) observed when \( x > 0 \). On the region of the space where \( x < 0 \), this loss might be arbitrarily large, but this has no effect on the loss incurred by any strategy that chooses this expert. Thus, the unconditional expectation, \( \mathbb{E}[L(\xi, z)] \), used in Section 2 to define the notion of sleeping regret and adopted in Section 3 for the independent case, is no longer relevant. In the dependent setting, only the conditional expectation of the loss given that the expert is awake should be considered when defining regret.

Formally, let \( A = \{A_1, \ldots, A_p\} \) be the set of all possible awake sets, and let \( A \) be the random variable that generates the i.i.d. sequence of awake sets \( \{A^t\}_{t=1}^{T} \) defined in Section 2. Then, the generalized sleeping regret \( \mathcal{R}_T^\text{SLEEP}(B) \) of an algorithm \( B \) is defined as follows:

\[
\mathcal{R}_T^\text{SLEEP}(B) = \sum_{t=1}^{T} \sum_{k=1}^{p} p_k \mathbb{E}[L(\xi_{i^*(k)}, z_t)] - \mathbb{E}[L(\xi_{i^*(k)}, z_t)|A^t = A_k],
\]

where \( p_k = \mathbb{P}[A^t = A_k] \) is the probability of the awake set \( A_k \), and \( i^*(k) = \min_{i \in A_k} \mathbb{E}[L(\xi_i, z)] \).

When awake sets and losses are independent, the generalized sleeping regret coincides with the notion of sleeping regret of Section 2 since, for any \( t \), the following holds:

\[
\mathbb{E}[L(\xi_{i_{i^*(k)}}, z_t)]
\]

\[
= \sum_{k=1}^{p} p_k \mathbb{E}[L(\xi_{i_{i^*(k)}}, z_t)|A^t = A_k]
\]

\[
= \sum_{k=1}^{p} p_k \mathbb{E}[L(\xi_{i_{i^*(k)}}, z_t)] = \sum_{k=1}^{p} p_k \min_{i \in A_k} \mathbb{E}[L(\xi_i, z_t)].
\]

When all experts are awake at each round, the generalized sleeping regret matches the standard definition of regret.

4.2. The UCB-SLG algorithm

In view of the discussion above, the simple strategy adopted in Auer-N, that is, averaging over the time steps where an expert was awake and where its loss was observed, cannot work here, since this would lead to arbitrarily biased empirical estimates. This is typically the case where experts can be awake only for certain regions of the input space \( X \). Thus, our main idea for tackling the dependency between losses and awake sets is to use empirical estimates conditioned on the awake sets and decompose the problem into \( p \) subproblems, one per awake set \( A_k, k \in [p] \). Our UCB-SLG
Our regret guarantee characterizes the benefit from the additional loss observations by partitioning the feedback graph \( G_k \) into cliques for each \( k \in [p] \) and taking the minimum over all such possible clique coverings. Specifically, we define a clique of a directed graph \( G_k \) as a set of vertices \( C \subset k \) that are all neighbors with each other, that is such that, for all \( i, j \in C \), we have \( i \in N_{k,j} \) and \( j \in N_{k,i} \). A clique covering \( C_k \) of graph \( G_k \) is defined as a set of cliques that satisfy \( \bigcup_{C \in C_k} C = \lambda_k \). In the following theorem, the minimum is over all sets of all clique coverings \( C_k \) for each graph \( G_k \) with \( k \in [p] \).

**Theorem 3** Assume that the sequence of awake sets \( A^t \) and loss values \( L(\xi_j, z_i), j \in [K], \) are generated jointly at time \( t \), but i.i.d. over time. Assume further that each awake set \( k \) admits a fixed feedback graph \( G_k \). Then, the generalized sleeping regret of the UCB-SLG algorithm after \( T \) rounds is bounded as follows:

\[
R_T^{\text{SLG}}(\text{UCB-SLG}) = O \left( \sum_{k=1}^{p} p_k \min_{C_k} \sum_{C \in C_k} \min_{j \in C \setminus B_k} \sum_{k,j} \frac{1}{(S_{k,j})^2} \log(T) \right).
\]

The generalized sleeping regret is decomposed into a sum over the regret for each awake set \( k \) times the probability of that awake set. If the probability \( p_k \) of an awake set \( k \) is very small, then the bound on the regret for this awake set is given less weight. As one would expect for UCB-type algorithms, the regret is logarithmic in \( T \) for each awake set \( k \), but, unlike standard bounds, the loss gap is based on conditional expectations. This makes this bound not readily comparable to that of Theorem 1, where awake sets and losses are independent, even in the bandit setting.

Our regret bound shows the benefit of using feedback graphs. In particular, if the graphs are denser, then there are more ways to partition the graphs. Thus, the \( \min_{C_k} \) is over a larger set, thereby potentially decreasing the overall bound. In other words, if more losses are revealed at each round, the bound on the regret decreases accordingly. In the case where there is a single region, \( p = 1 \), with one fixed graph \( G^1 \), the bound reduces to that of the setting analyzed in Caron et al. (2012), and our regret bound for UCB-SLG matches that of the UCB-N algorithm therein.

## 5. Online learning with abstention

In this section, we apply the ideas introduced in Section 4 to the setting of online learning with abstention.
Online Learning with Sleeping Experts and Feedback Graphs

Online learning with abstention is a scenario recently introduced by Cortes et al. (2018), where a learner can elect to abstain from making a prediction at the price of a certain cost \( c > 0 \). When the learner abstains, she does not receive the label of the current input. The benefit of abstaining is that the loss incurred, \( c \), is typically lower than the loss of incorrectly predicting a label.

While Cortes et al. (2018) cast the online abstention setting as an instance of online learning with feedback graphs, in this section, we cast the problem as an instance of online learning with both feedback graphs and sleeping experts. As we shall see, this choice provides us with a more meaningful and challenging benchmark for the learner, as well as an algorithm that achieves sublinear regret with respect to this benchmark. As a result, the algorithms we present achieve a more favorable empirical performance than those presented in Cortes et al. (2018), outperforming even an unrealistic full information algorithm designed only for the online abstention (but not feedback graph) setting.

We will adopt the notation of Cortes et al. (2018): let \( r : \mathcal{X} \to \mathbb{R} \) denote an abstention function that dictates which examples to abstain on, and let \( h : \mathcal{X} \to \mathbb{R} \) denote a prediction function that determines the predicted labels of the examples. Let \( \mathcal{E} = \{ \xi_j = (h_j, r_j) : j \in [K] \} \subset \mathcal{H} \times \mathcal{R} \) denote a family of experts made up of pairs of a prediction function in \( \mathcal{H} \) and an abstention function in \( \mathcal{R} \).

The online abstention protocol is as follows. At each round \( t \in [T] \), the learner receives an input point \( x_t \in \mathcal{X} \) drawn i.i.d. according to the marginal distribution associated with \( \mathcal{D} \), and chooses an index \( I_t \in [K] \) corresponding to a pair \( \xi_{I_t} = (h_{I_t}, r_{I_t}) \). The learner determines whether to make a prediction based on the value of the abstention function, \( r_{I_t}(x_t) \). If \( r_{I_t}(x_t) \leq 0 \), the learner abstains and incurs a fixed loss \( c \in \mathbb{R}_+ \). If \( r_{I_t}(x_t) > 0 \), the learner predicts, her prediction being \( h_{I_t}(x_t) \). In this case (and only in this case), she receives a label \( y_t \in \{ \pm 1 \} \), and incurs the prediction loss \( \ell(y_t, h_{I_t}(x_t)) \). One natural choice for an abstention function \( r \) associated with a prediction function \( h \) is a confidence-based function measuring the magnitude of \( h \), that is \( r(x) = |h(x)| - \gamma \) for some threshold \( \gamma \geq 0 \) (e.g., Bartlett & Wegkamp, 2008). In the sequel, we focus on the binary classification problem where \( \ell(y, h(x)) \in \{0, 1\} \) can be the 0/1 loss, \( \mathbb{I}\{y h(x) \leq 0\} \) or any of its bounded surrogates. The abstention loss of the pair \( \xi = (h, r) \) on example \( (x, y) \in \mathcal{X} \times \{ \pm 1 \} \) is defined as

\[
L(\xi, z) = \ell(y, h(x))\mathbb{I}\{r(x) > 0\} + c\mathbb{I}\{r(x) \leq 0\}.
\]

At first glance, the online abstention and sleeping expert settings appear to be different frameworks. Yet, we can cast the online abstention setting as a variant of the sleeping experts one by carefully defining an awake set that captures the loss

---

\(^{2}\mathrm{Here}, \mathbb{I}\{ \cdot \} \text{ denotes the indicator function.} \)
Algorithm 3: UCB-ABS.

Init: \( O_k(0) = 1 \) for all \( k \in [p] \);

for \( t \geq 1 \) do

\[
S_k(t) \leftarrow \sqrt{\frac{5 \log(t)}{O_k(t-1)}} \quad \text{for all } k \in [p];
\]

Receive \( x_t \in \mathcal{X} \), and awake set \( A^t \subseteq \mathcal{E} \);

Let \( k \in [p] \) be such that \( A^t = A_k \);

Receive graph \( G_k \) with out-neighbors \( N_{k,i}, i \in A_k \);

if \( \min_{i \in A_k \setminus \{0\}} \hat{\nu}_{k,i}(t-1) - S_k(t-1) < c \) then

\[
I_t \leftarrow \underset{i \in A_k \setminus \{0\}}{\text{argmin}} \hat{\nu}_{k,i}(t-1);
\]

Reveal \( y_t \);

\( O_k(t) \leftarrow O_k(t-1) + 1; \)

for \( j \in A_k \setminus \{0\} \) do

\[
\hat{\nu}_{k,j}(t) \leftarrow \frac{\hat{\nu}_{k,j}(x_t)}{O_k(t)} + \left(1 - \frac{1}{O_k(t)}\right) \hat{\nu}_{k,j}(t-1);
\]

end for

else

\( I_t \leftarrow 0; \)

end if

end for

The special property of the all-abstain expert.

Given the event \( A^t = A_k \), the UCB-ABS algorithm chooses the all-abstain expert if \( c \) is less than the smallest lowest confidence bound of \( \nu_{k,i} \) over the experts \( \xi_i \neq \xi_0 \) in the awake set (i.e., all \( i \in A_k \setminus \{0\} \)). If this is not the case, then it chooses the expert with index \( i \in A_k \setminus \{0\} \) having the smallest estimated conditional loss. In short, if \( A^t = A_k \), the UCB-ABS algorithm picks the expert with index:

\[
I_t = \begin{cases} 
0 & \text{if } c < \min_{i \in A_k \setminus \{0\}} \hat{\nu}_{k,i}(t-1) - S_k(t-1) \text{ and } \arg\min_{i \in A_k \setminus \{0\}} \hat{\nu}_{k,i}(t-1) \text{ otherwise.} 
\end{cases}
\]

Algorithm 3 shows the pseudocode. Notice that, unlike the previous section, there is no need to maintain individual statistics for each expert here. In particular, the quantities \( S_k(t) \) and \( O_k(t) \) now refer to \( A_k \). This is due to the specific structure of the feedback graphs \( G_k \), as explained below.

The losses revealed at each round depend on whether the chosen expert is the all-abstain expert. For each awake set \( A_k \), suppose that expert \( \xi_i \) is not the all-abstain expert. Then if this expert is chosen at time \( t \), the true label \( y_t \) is revealed, so the loss of all experts is observed. On the other hand, if the all-abstain expert is chosen, then only the loss of the all-abstain expert is revealed. Thus, for each awake set \( A_k \), there is one fixed graph \( G_k = (A_k, E_k) \) whose out-neighbors are defined as follows: \( N_{k,i} = A_k \) if \( i \neq 0 \) and \( N_{k,i} = \{0\} \) if \( i = 0 \). See Figure 2 for an illustration of the graph \( G_k \). This feedback graph \( G_k \) is, in fact, the largest feedback graph per awake set that can be constructed in the abstention setting. Notice that after we condition over each set \( A^t = A_k \), this algorithm is almost running Follow-The-Leader in the full information setting on the experts of the awake set (excluding the all-abstain expert).

The computational complexity of the algorithm depends on \( p \) since it keeps estimates of the conditional losses for each awake set in \( A_k, k \in [p] \). Since the awake sets are the intersections of the accepting regions, one can define abstention functions \( \tau_i \) such that the resulting number of awake sets \( p \) is not too large. For example, in the scenario where the hypothesis functions \( h_i \) perform well in complementary regions of the input space, we can define abstention functions whose non-abstention regions do not overlap in such a way that \( p = K \). This is conceivable, for example, in a recommendation system setting, where regions correspond to
general categories of an item and some hypothesis functions might be better at making recommendations within a certain category and hence should only be awake for that category.

6. Experiments

In this section, we present the results of several experiments for the online learning with abstention setting described in Section 5. These experiments demonstrate that UCB-ABS admits a strong empirical performance.

We compare UCB-ABS to several baselines, including the algorithms UCB-GT, UCB-NT, and FS of Cortes et al. (2018), as well as standard UCB (Auer et al., 2002). FS is the ideal comparator that picks the expert with the smallest empirical mean, but has the advantage that the losses of all the experts are revealed at each round. Thus FS and UCB lie at the two ends of a spectrum. On one end, the FS algorithm has access to the full loss information at each round, at the other end, the UCB algorithm only sees the loss of the expert chosen. In between these two extremes, the losses revealed to UCB-GT, UCB-NT, and UCB-ABS depend on whether the chosen expert for each algorithm abstains or predicts. That is, if the chosen expert predicts at time $t$, the true label $y_t$ is revealed and hence the losses of all the experts are observed, while if it abstains at time $t$, only the loss of the abstaining experts, which is simply equal to the abstention cost $c$, is revealed. It is important to note that UCB-ABS uses all the revealed expert losses to update its empirical estimates at each round, while UCB-GT and UCB-NT only use a subset of the revealed losses at each round, since these latter algorithms can only make updates based on information up to time $t - 1$. On the other hand, FS relies on full information no matter what, and is therefore relying on informational assumptions which are clearly outside the abstention setting.

For ease of comparison, in our experiments, we adopted the same setup and used the same datasets as Cortes et al. (2018). That is, the predictions functions $h$ are random hyperplanes centered at the origin with normal vectors drawn randomly from the Gaussian distribution $N(0, 1)^d$ where $d$ is the feature dimension, and the abstention functions $r$ are concentric annuli centered at the origin with radii in $(0, \frac{\sqrt{d}}{20}, \frac{2\sqrt{d}}{20}, \ldots, \sqrt{d})$. We tested abstention costs $c$ in $\{0.05, 0.1, 0.15, \ldots, 0.9\}$. We used the CIFAR dataset from Krizhevsky et al. (2009), where we extracted the first twenty-five principal components, and used eight UCI datasets: HIGGS, phishing, ijcnn, covtype, eye, skin, cod-rna, and guide. The loss of each algorithm was calculated as follows: First we fixed the set of experts and averaged the results over five random draws of the data, and then let the experts vary and averaged the results over five random draws of the experts.

Figure 3 shows the averaged abstention loss $L(.)/t$ with standard deviations for the different abstention costs. In Appendix E, we show the plots of all the datasets we tested, where the same patterns recur. These experiments show that UCB-ABS outperforms UCB-NT and UCB on all datasets and it attains a better averaged loss than that of UCB-GT on most datasets. Remarkably, on some datasets UCB-ABS even outperforms FS, which is an unrealistic baseline that clearly violates the rules of the abstention setup in that this algorithm receives all loss information at each round. This algorithm was used for its ideal performance in Cortes et al. (2018). Thus, thanks to a generalized notion of sleeping regret and casting the abstention problem as an instance of the sleeping experts framework, we obtain both theoretical and empirical improvements. In addition to the experiments above, we tested the effects of increasing the number of abstention and prediction functions. We also present plots for the fraction of points each algorithm abstains on – see Appendix E.

7. Conclusion

We presented a comprehensive analysis of online learning with sleeping experts and feedback graphs, combining two lines of existing work that are closely related but have so far not been considered together. We presented both algorithmic solutions and theoretical analysis, and we also adapted our ideas to the online abstention problem, with extensive experiments showing that our adaptation outperforms existing solutions. While our experiments focused on binary classification, they can be directly extended to multiclass classification and regression problems.
Acknowledgments

This work was partly funded by NSF CCF-1535987 and NSF IIS-1618662. The authors would also like to thank Ningshan Zhang for helpful feedback during this work.

References


