Online Learning with Dependent Stochastic Feedback Graphs

Corinna Cortes [1] **Giulia DeSalvo** [1] **Claudio Gentile** [1] **Mehryar Mohri** [1] **Ningshan Zhang** [2]

Abstract

A general framework for online learning with partial information is one where feedback graphs specify which losses can be observed by the learner. We study a challenging scenario where feedback graphs vary stochastically with time and, more importantly, where graphs and losses are dependent. This scenario appears in several real-world applications that we describe where the outcome of actions are correlated. We devise a new algorithm for this setting that exploits the stochastic properties of the graphs and that benefits from favorable regret guarantees. We present a detailed theoretical analysis of this algorithm, and also report the results of a series of experiments on real-world datasets, which show that our algorithm outperforms standard baselines for online learning with feedback graphs.

1. Introduction

Prediction with expert advice is a classical framework for modeling the repeated interactions between a learner and the environment (Littlestone & Warmuth, 1994; Cesa-Bianchi & Lugosi, 2006). In this framework, the learner maintains performance estimates for a set of experts (or actions), based on their past behavior. At each round, she uses those estimates to select an action, which incurs a loss determined by the environment, and subsequently updates her estimates. The learner's objective is to minimize her regret, that is the difference between her cumulative loss over a finite number of interactions and that of the best expert in hindsight.

The two most standard settings in this framework are the *full information* setting, where the learner is informed of the loss incurred by all actions, and the *bandit* setting, where she can only observe the loss incurred by the selected action. These settings are both special instances of a general model of online learning introduced by Mannor & Shamir (2011), where loss observability is specified by a *feedback graph*. In a directed feedback graph, each vertex represents an action and an edge from vertex i to vertex j indicates that the loss of action j is observed when action i is selected. The bandit setting corresponds to a feedback graph reduced to only self-loops at each vertex, the full information setting corresponds to a fully connected graph. Online learning with feedback graphs has been further extensively analyzed by Alon et al. (2013; 2017) and by a series of other recent publications (Caron et al., 2012; Alon et al., 2015; Kocák et al., 2014; Neu, 2015; Cohen et al., 2016; Buccapatnam et al., 2014b; Wu et al., 2015a; Tossou et al., 2017; Liu et al., 2018; Yun et al., 2018; Cortes et al., 2019; Li et al., 2020).

In the scenarios of online learning with feedback graphs studied by some authors, the feedback graph G is assumed to be fixed over time (Caron et al., 2012; Buccapatnam et al., 2014b; Neu, 2015), while others allow a time-varying feedback graph G_t, chosen adversarially at each round t (Alon et al., 2013; 2017; 2015; Kocák et al., 2014; Cohen et al., 2016; Kocák et al., 2016). This paper considers a scenario commonly appearing in practice where feedback graphs are time-varying but are *stochastic*, that is, at each round t, the feedback graph G_t is drawn i.i.d. according to some unknown distribution. Crucially, we will not assume that the losses and feedback graphs are statistically independent, a key assumption adopted by many authors, including Cohen et al. (2016); Tossou et al. (2017); Liu et al. (2018); Li et al. (2020), in their analysis of (partially accessible) feedback graphs. The novelty of the setting presented in this paper is thus not just to study stochastic feedback graphs, but more importantly, to allow for an underlying dependence between the feedback graphs and the losses, which as we will see makes the analysis intrinsically harder. We call this new setting *online learning with dependent stochastic feedback graphs* highlighting that losses and graphs are dependent stochastic variables.

Our scenario of online learning with dependent stochastic graphs is relevant to many real-world problems. As an example, consider the problem of a doctor selecting an appropriate treatment (action) for a patient, based on his symptoms. The patient can be modeled by a feature vector x_t representing his health history and other characteristics and we assume a fixed distribution over the patients x_t

[*]Equal contribution [1]Google Research, New York, NY [2]Hudson River Trading, New York, NY. Correspondence to: Giulia DeSalvo <giuliad@google.com>.

Proceedings of the 37^{th} International Conference on Machine Learning, Online, PMLR 119, 2020. Copyright 2020 by the author(s).

visiting that doctor. Each x_t induces a feedback graph over treatments, since the success of treatment i for curing a collection of symptoms that the patient displays could also reveal the effectiveness of an alternate treatment j. Thus, just as x_t, the feedback graph is stochastic and since both the loss and the feedback graph depend on x_t, they are *not* independent. More generally, the scenario of online learning with stochastic feedback graphs is pertinent to a variety of problems where a stochastic context or feature vector x_t induces a relationship between experts or actions, as the outcome of one expert for x_t also reveals information about that of some other experts. We will discuss in more detail several examples of such structured experts.

Our study admits some connection with that of Kocák et al. (2016), who examine a scenario of online learning with an *imperfect feedback*, modeled by a weighted graph, or the closely related one of Wu et al. (2015b); however, the definition of the edge weights in those studies is entirely distinct from ours. Related to feedback graph problems, but in settings quite different from ours, Yun et al. (2018) consider stochastic bandits where the extra information admits a cost to be traded off against regret, while Cortes et al. (2019) analyze feedback graphs in the context of the sleeping experts.

Our setting naturally raises the following question: how can we leverage the stochastic properties of the graphs while at the same time dealing with the dependency between losses and graphs? One can prove that simply averaging loss observations of an expert based on the feedback graph G_t may result in an arbitrarily biased empirical estimate of the expected loss of the expert, since there is a dependency between losses and graphs and since we make no assumptions on the nature of this dependency. In order to avoid this pitfall, we introduce an *estimated feedback graph* $\widehat{G}_t$ based on past observations up to time $t-1$ instead of relying on the feedback graph G_t at the time t. The design of our algorithms then relies on the observability specified by $\widehat{G}_t$ so that the empirical estimates defined by averaging loss observations based on the estimated graphs will be unbiased. Specifically, the empirical estimate of the performance of action j is updated whenever action, I_t, is selected by the algorithm and there is an edge from I_t to j in $\widehat{G}_t$.

Using estimated graphs, however, leads to another complication: there are times t when even though there is an edge from I_t to j in $\widehat{G}_t$, the loss of action j is not revealed, which is when graph G_t does not contain an edge from I_t to j. In order for these events not to occur too often, we introduce a *threshold* θ_t to remove edges with low probability in the estimated graph. That is, depending only on information up to time $t-1$, we estimate the probability of an edge from vertex i to vertex j and define, at each round t, the graph $\widehat{G}_t$ whose edges admit a probability above this threshold θ_t. Furthermore, over these events, since the loss of action j is not revealed, we also introduce an *estimated loss* used to update the empirical estimate of the performance of expert j. These estimated losses effectively estimate the expectation of the loss of expert j when there is no edge from i to j in graph G_t.

Intuitively, denser feedback graphs imply that more loss observations are revealed and hence, the algorithm should converge faster since the empirical estimates more quickly converge to their respective means. The novelty of our approach is thus based on carefully leveraging the interplay between the estimated graphs, thresholds, and estimated losses in order to reliably update the empirical estimates of the expected loss of an expert via the densest graph possible, thereby circumventing the dependency of losses and graphs while also exploiting the presence of edges in the feedback graph in order to attain favorable theoretical guarantees.

We start with a formal description of the scenario of online learning with dependent stochastic feedback graphs and introduce the relevant notation (Section 2). In Section 3, we present a novel algorithm that exploits the stochasticity of the feedback graphs and carefully deals with the dependency between losses and graphs. In Section 4, we prove that our algorithm benefits from favorable pseudo-regret guarantees that are logarithmic in the number of rounds T and are expressed in terms of the expected feedback graph, $\mathbb{E}[G_t]$. In that section, we also illustrate, through a lower bounding argument, that the dependency structure between losses and graphs may render the extra side information delivered by feedback graphs mostly useless. In Section 5, we discuss a natural scenario with stochastic graphs and report the results of several experiments on real-world datasets suggesting that our algorithm outperforms standard baselines.

2. Learning Scenario

We consider the familiar sequential learning problem over T rounds, where the algorithm has access to a set of K experts (or actions) $\mathcal{E} = \{\xi_1, \ldots, \xi_K\}$, which we abusively identify as $\mathcal{E} = \{1, \ldots, K\}$ in some cases. At each round $t \in [T]$, each expert $\xi_i \in \mathcal{E}$ is assigned a loss $\ell_t(\xi_i) \in [0,1]$. The algorithm selects an expert ξ_{I_t} and incurs the corresponding loss $\ell_t(\xi_{I_t})$. Additionally, at round $t \in [T]$, the algorithm is supplied with a feedback graph $G_t = (\mathcal{E}, E_t)$, where each vertex corresponds to an expert, and where the presence of an edge from vertex i to vertex j indicates that the loss of expert ξ_j is revealed to the algorithm if it selects expert ξ_i at time t. In what follows, we denote by $N_t(i)$ the out-neighborhood of vertex i in G_t, that is the set of vertices j that can be reached from i via an edge in E_t. Note, the feedback graph G_t only specifies the observability of expert losses and not loss values.

We assume, in what follows, that the graphs G_t admit self-loops at all vertices, that is $i \in N_t(i)$ for all $i \in \mathcal{E}$. This guarantees that the information received by the algorithm includes at least the loss value $\ell_t(\xi_{I_t})$ of the action ξ_{I_t} it selects, that is the information available in the bandit setting.

We consider the setting where expert losses and feedback graphs are stochastic and *arbitrarily* dependent. Let $\mathcal{G}$ denote the family of all directed graphs with K vertices (and self-loops on all of them), and let $\mathcal{D}$ be a distribution over $\mathcal{G} \times [0,1]^K$ unknown to the algorithm. At each round $t \in [T]$, the environment generates the pair $(G_t, \overline{\ell}_t) \in \mathcal{G} \times [0,1]^K$ i.i.d. according to $\mathcal{D}$, where $\overline{\ell}_t = (\ell_t(\xi_1), \ldots, \ell_t(\xi_K))$ is the vector of all K expert losses.

In particular, this setting includes the important learning scenario of statistical learning, so far not studied in the bandits literature, where pairs (x_t, y_t) are drawn i.i.d. according to some distribution over $\mathcal{X} \times \mathcal{Y}$, where $\mathcal{X}$ is the input space and $\mathcal{Y}$ is the output space. Expert ξ_i is a mapping from $\mathcal{X}$ to $\mathcal{Y}$ and its loss at time t is given by $\ell_t(\xi_i) = L(\xi_i(x_t), y_t)$, for some loss function L. The feedback graph G_t is then typically a function of x_t. As an example, consider the following sequence of graphs: at each time t, graph G_t admits (bi-directional) edges (i,j) whenever $\xi_i(x_t) = \xi_j(x_t)$. The probability $p_{i,j}$ of an edge in this graph is then given by $p_{i,j} = \mathbb{P}(\{x \colon \xi_i(x) = \xi_j(x)\})$, where $p_{i,i} = 1$ if the graph admits self-loops. Experts may be, for example, decision trees with the same function value in a given region of the space, which thereby implies that these trees will admit the same loss in this region, and the graph G_t will then account for the shared loss observability between these trees whenever an x_t falls in this region. In such scenarios, G_t and the losses $\overline{\ell}_t$ are dependent as they are both functions of x_t. Note that the full vector of losses $\overline{\ell}_t$ is generated i.i.d. at each time t, but its components $\ell_t(\xi_i)$ can be dependent. Likewise, the graphs G_t are i.i.d. across time but, for any given t, their edges (i,j) may be correlated.

Another instance of this scenario is *learning with abstention*, where an expert can elect to either make a prediction or abstain, at a cost typically less than that of misclassification (Cortes et al., 2016; 2018). In the stochastic case, both the loss vector and the feedback graph at round t are functions of x_t with the feedback graph G_t defined by the following: if the expert selected by the algorithm elects to predict on input x_t, then the label y_t is revealed, and the loss $\ell_t(\xi_i) = L(\xi_i(x_t), y_t)$ of every expert ξ_i is revealed; otherwise, ξ_{I_t} abstains, the label y_t is not revealed and only the loss of ξ_{I_t} along with that of other abstaining experts is revealed, which is some fixed abstention cost c.

We consider the so-called *uninformed case*, where graph G_t is revealed to the algorithm only after it has selected an action to play. The standard measure of the performance of the algorithm is the *pseudo-regret*, R_T, defined as follows:

$$R_T = \max_{\xi \in \mathcal{E}} \mathbb{E}\left[\sum_{t=1}^{T} \ell_t(\xi_{I_t}) - \ell_t(\xi)\right],$$

where the expectation is taken over the i.i.d. sequence of pairs $(G_t, \overline{\ell}_t)$ drawn from $\mathcal{D}$.

3. Algorithm

In this section, we present our new algorithm UCB-DSG, or UCB *with Dependent Stochastic Graphs*, for the stochastic and loss-dependent feedback graph scenario just discussed. In Section 4, we prove that this algorithm admits favorable pseudo-regret guarantees. The design of UCB-DSG builds on the classical UCB algorithm of Auer et al. (2002a). The pseudocode of UCB-DSG is given in Algorithm 1.

At a high level, UCB-DSG maintains an empirical estimate $\widehat{\mu}_{i,t}$ for the average loss $\mu_i = \mathbb{E}[\ell(\xi_i)]$ of each expert $\xi_i \in \mathcal{E}$. At each round, it selects the expert with the smallest lower confidence estimate, and uses the loss observations from a feedback graph, defined below, to update its estimates.

Ideally, the algorithm would update its empirical means at round t according to graph G_t. However, since G_t may not be a complete graph and may depend on the losses $\ell_t(\xi_i)$, simply taking an average over the losses observed could lead to biased empirical estimates. As a straightforward example, consider a graph G_t that, when selecting expert ξ_i, also reveals the loss of expert ξ_j, but only when $\ell_t(\xi_j) \geqslant 0.5$. For a detailed discussion of this technical bias problem in the context of the online learning with abstention, see (Cortes et al., 2018). Section 4 below contains an illustrative example in the form of a lower bound that shows that even if the (marginal) probabilities $p_{i,j}$ of edges (i,j) are all close to 1, we cannot in general take advantage of the presence of these edges when graphs and losses are statistically dependent.

The first key idea behind the design of our algorithm is to use at time t a surrogate graph $\widehat{G}_t$ derived from an average of past observed graphs $G_1, \ldots, G_{t-1}$, which bypasses this technical bias issue. However, the loss of an expert ξ_j in the out-neighborhood $\widehat{N}_t(I_t)$ of graph $\widehat{G}_t$ may not be actually observed at round t. This is because, in general, graphs G_t and $\widehat{G}_t$ do not coincide and $N_t(I_t)$ may not contain ξ_j. In order to control the estimation error from such cases, we introduce a threshold that is used to trim the estimated graph of any low-probability edges. Moreover, in these cases, UCB-DSG resorts to an *estimated loss* $\widetilde{\ell}_t(\xi_j)$ for expert ξ_j since the true loss is not revealed. The threshold and the estimated losses are carefully crafted to guarantee that the true mean of each expert is, with high probability, within a confidence interval around the empirical mean. Specifically, there is a critical interplay between the estimated losses and thresholds to favor a denser graph while at the same time

ALGORITHM 1: UCB-DSG

Parameters: Experts $\mathcal{E} = \{\xi_1, \dots, \xi_K\}$, # of rounds T;
Init: $U_{i,0} = Q_{i,0} = 1$ for all $i \in [K]$;
for $t \in [T]$ **do**
  $S_{i,t-1} \leftarrow \frac{O(\log(Kt))}{\sqrt{Q_{i,t-1}}}$; {Constants in App.B}
  $I_t \leftarrow \operatorname{argmin}_{i\in[K]}(\widehat{\mu}_{i,t-1} - S_{i,t-1})$;
  $\widehat{N}_t(I_t) \leftarrow \{j \in [K] \colon \widehat{p}_{I_t,j}^{t-1} \geqslant \theta_{j,t-1}\}$; {Create $\widehat{G}_t$}
  for $j \in [K]$ **do**
    if $j \in \widehat{N}_t(I_t)$ **then**
      $Q_{j,t} \leftarrow Q_{j,t-1} + 1$;
      if $j \in N_t(I_t)$ **then**
        $\widehat{\mu}_{j,t} \leftarrow \frac{\ell_t(\xi_j)}{Q_{j,t}} + (1 - \frac{1}{Q_{j,t}})\widehat{\mu}_{j,t-1}$;
        {Use true loss}
      else
        $\widehat{\mu}_{j,t} \leftarrow \frac{\widetilde{\ell}_t(\xi_j)}{Q_{j,t}} + (1 - \frac{1}{Q_{j,t}})\widehat{\mu}_{j,t-1}$;
        {Use estimated loss}
    else
      $Q_{j,t} \leftarrow Q_{j,t-1}; S_{j,t} \leftarrow S_{j,t-1}; \widehat{\mu}_{j,t} \leftarrow \widehat{\mu}_{j,t-1}$;
      {No update}
    if $\widehat{p}_{I_t,j}^{t-1} \geqslant 1 - \frac{O(\log(Kt))+1}{\sqrt{U_{j,t-1}+1}}$ **then**
      $U_{j,t} \leftarrow U_{j,t-1} + 1$;
      for $i \in [K]$ **do**
        $\widetilde{\nu}_{j,i,t} \leftarrow \frac{\ell_t(\xi_j)\mathbb{I}\{j\notin N_t(i), j\in N_t(I_t)\}}{U_{j,t}} + \left(1 - \frac{1}{U_{j,t}}\right)\widetilde{\nu}_{j,i,t-1}$;
    else
      $U_{j,t} \leftarrow U_{j,t-1}; \widetilde{\nu}_{j,i,t} \leftarrow \widetilde{\nu}_{j,i,t-1}, \forall i \in [K]$;
      {No update} .

reliably updating the empirical estimates, $\widehat{\mu}_{i,t}$.

To devise our surrogate graph $\widehat{G}_t$, for $t \geqslant 2$ and for all $i, j \in [K]$, we use an unbiased estimate of the probability of a directed edge from i to j, $\widehat{p}_{i,j}^{t-1}$, based on the information available up to time $t-1$:

$$\widehat{p}_{i,j}^{t-1} = \frac{\sum_{s=1}^{t-1} \mathbb{I}\{j \in N_s(i)\}}{t-1} ,$$

where $\mathbb{I}\{\cdot\}$ denotes the indicator function of the argument, and $\widehat{p}_{i,j}^{0} = 0$. We define $\widehat{G}_t$ by allowing an edge from i to j only if $\widehat{p}_{i,j}^{t-1}$ is above the aforementioned threshold. That is, the graph $\widehat{G}_t$ is determined by the out-neighborhoods $\widehat{N}_t(i) = \left\{j \in [K] \colon \widehat{p}_{i,j}^{t-1} \geqslant \theta_{j,t-1}\right\} \cup \{i\}$, with the threshold $\theta_{j,t-1}$ specified by

$$\theta_{j,t-1} = \min\left\{2 - 2\sqrt{\frac{U_{j,t-1}+1}{Q_{j,t-1}+1}} - \frac{O(\log(Kt))}{\sqrt{Q_{j,t-1}+1}}, 1 - \frac{O(\log(Kt))}{\sqrt{Q_{j,t-1}+1}}\right\} . \tag{1}$$

In the above, $Q_{j,t-1}$ is the number of times expert ξ_j was updated up to time $t-1$ and similarly, $U_{j,t-1}$ is the number of times the estimated loss, $\widetilde{\ell}_t(\xi_j)$, has been updated up to time $t-1$. The exact constants in front of the log terms hidden in the big-O notation are given in Appendix B. Note that, by construction, $\widehat{G}_t$ admits self-loops at all vertices.

Since the algorithm uses the estimated loss of expert ξ_j when the loss of expert ξ_j is not revealed, that is when $j \notin N_t(I_t)$, our estimated loss aims at estimating the following conditional expected loss:

$$\mathbb{E}[\ell_t(\xi_j) | j \notin N_t(I_t)] = \frac{\mathbb{E}[\ell_t(\xi_j)\mathbb{I}\{j \notin N_t(I_t)\}]}{\mathbb{E}[\mathbb{I}\{j \notin N_t(I_t)\}]} .$$

The denominator in the above fraction can be readily estimated by $1 - \widehat{p}_{I_t,j}^{t-1}$, which converges to its (conditional) mean, $1 - p_{I_t,j}$. Concretely, for each pair $i, j \in K \times K$ and time $t \in [T]$, we maintain estimates $\widehat{p}_{i,j}^{t-1}$ and, at time t, the probability estimate $\widehat{p}_{i,j}^{t-1}$ corresponding to $i = I_t$ is used.

On the other hand for the numerator $\mathbb{E}[\ell(\xi_j)\mathbb{I}\{j \notin N(i)\}]$, we need to introduce a new estimator and so we define

$$\widetilde{\nu}_{j,i,t} = \frac{1}{U_{j,t}} \sum_{s=1}^{t} \ell_s(\xi_j)\mathbb{I}\{j \notin N_s(i)\}\mathbb{I}\{j \in N_s(I_s)\}C_{j,s}$$

with

$$C_{j,s} = \mathbb{I}\left\{\widehat{p}_{I_s,j}^{s-1} \geqslant 1 - \frac{O(\log(Ks))+1}{\sqrt{U_{j,s-1}+1}}\right\} \text{ and } U_{j,t} = \sum_{s=1}^{t} C_{j,s}$$

for all j, i, and t. Note that the sum in the definition of $\widetilde{\nu}_{j,i,t}$ must be over the times s such that $j \in N_s(I_s)$ since the loss of expert ξ_j is revealed only when playing action I_s. A careful reader would notice that this injects a bias in the estimator $\widetilde{\nu}_{j,i,t}$, but the condition defining $C_{j,s}$, which holds whenever the (estimated) probability of the edge from I_s to j is large enough, is chosen so as to guarantee that $\widetilde{\nu}_{j,i,t}$ converges to $\mathbb{E}[\ell(\xi_j)\mathbb{I}\{j \notin N(i)\}]$. One may, in fact, adopt a condition like $C_{j,s}$ in order to define the estimated graph $\widehat{G}_s$ directly, but this would result in a worse regret guarantee, since $C_{j,s}$ is provably more conservative than the condition $\widehat{p}_{I_s,j}^{s-1} \geqslant \theta_{j,s-1}$ used in $\widehat{G}_s$. Putting the above together, our estimated loss is defined as

$$\widetilde{\ell}_t(\xi_j) = \min\left\{\frac{\widetilde{\nu}_{j,I_t,t-1}}{1 - \widehat{p}_{I_t,j}^{t-1}}, 1\right\} .$$

Thresholds $\theta_{j,t}$ and estimated losses $\widetilde{\ell}_t(\xi_j)$ together play a key role in the regret guarantee of our algorithm. As we will prove in our analysis, a small threshold $\theta_{j,t}$ (implying that the algorithm will resort to estimated losses more often) results in a more favorable regret bound. The value of $\theta_{j,t}$ in (1) chiefly depends on the interplay between the two quantities $U_{j,t}$ and $Q_{j,t}$. One can show by induction

(see Appendix B) that $U_{j,t} \leqslant Q_{j,t} + 1$ for all j and t. Hence, if $U_{j,t}$ is as large as $Q_{j,t}$ (which implies that $\widetilde{\ell}_t(\xi_j)$ is a very accurate estimator of its own expectation), then $\theta_{j,t}$ will be small. On the other hand, when $U_{j,t}$ is much smaller than $Q_{j,t}$, then the estimated loss $\widetilde{\ell}_t(\xi_j)$ is a less reliable estimator. In this case, the threshold is $\theta_{j,t} = 1 - 2\frac{O(\log(Kt))}{\sqrt{Q_{j,t}+1}}$, which will be large if action j has already undergone many updates. In particular, if the algorithm has already resorted to $\widetilde{\ell}_t(\xi_j)$ very often (and $\widetilde{\ell}_t(\xi_j)$ is not dependable because of a small $U_{j,t}$), a large $\theta_{j,t}$ will ensure that the algorithm will rely on $\widetilde{\ell}_t(\xi_j)$ less frequently in the future. Finally, $\theta_{j,t}$ also mildly decreases with time, due to its logarithmic dependence on t through $O(\log(Kt))$.

In summary, at each round $t \in [T]$, UCB-DSG selects the expert with the smallest optimistic empirical loss, that is, $I_t = \operatorname{argmin}_{i\in[K]} \widehat{\mu}_{i,t-1} - S_{i,t-1}$, and operates with the surrogate graph $\widehat{G}_t$ to update all estimates $\widehat{\mu}_{j,t}$, for $j \in \widehat{N}_t(I_t)$, by either using a true loss $\ell_t(\xi_j)$, when available ($j \in N_t(I_t)$), or a estimated loss $\widetilde{\ell}_t(\xi_j)$ otherwise. Notice that, since the definition of $\widehat{G}_t$ only depends on graphs $G_1, \ldots, G_{t-1}$ and not on G_t, Algorithm 1 need not receive graph G_t before playing action I_t (uninformed setting).

4. Theoretical Guarantees

In this section, we analyze the UCB-DSG algorithm just described and prove that it benefits from favorable logarithmic pseudo-regret guarantees in terms of the independent sets of certain graphs derived from the matrix of probabilities $[p_{i,j}]_{i,j=1}^{K\times K}$. Our regret upper bound is given below, a proof sketch is provided in Section 4.1 and detailed in Appendix B. In Section 4.2, we complement our upper bound with a lower bound showing that in the case of strong dependency between graphs and losses, even when $p_{i,j}$ is close to 1 for all i and j, we cannot in general achieve better regret performance than $O(K)$.

As in standard UCB pseudo-regret bounds, our guarantees are expressed in terms of the loss gaps $\Delta_j = \mu_j - \mu_{i^*}$, $j \in [K]$, where μ_j is the expected loss of ξ_j and $\mu_{i^*} = \min_{j\in[K]} \mu_j$ that of the best expert, ξ_{i^*}. Naturally, our results are also expressed in terms of graph properties. For any $i, j \in [K]$, let $p_{i,j}$ denote the marginal probability of an edge from i to j. We define G as the weighted directed graph that admits an edge from i to j with weight $p_{i,j}$, for any $i, j \in [K]$. G can then be viewed as the *process graph* from which graphs G_t are drawn i.i.d., or their expectation, $\mathbb{E}[G_t]$. For any vector of thresholds $\bar{\theta} = \{\theta_1, \ldots, \theta_K\} \in [0, 1]^K$, we define $G_{\bar{\theta}} = \{\mathcal{E}, E_{\bar{\theta}}\}$ as the unweighted and undirected graph derived from G by keeping only those edges in G that satisfy both $p_{i,j} \geqslant \theta_j$ and $p_{j,i} \geqslant \theta_i$. In our bound, we only consider strictly positive loss gaps, $\Delta_j > 0$ and omit this condition to avoid a notational burden. Recalling that an independent set, $I(G)$, of an undirected graph G is the set of vertices such that no two vertices in $I(G)$ are adjacent, we denote by $\mathcal{I}(G) = \{I^1(G), I^2(G), \ldots\}$ the set of all independent sets of graph G. The bound in the theorem that follows is in terms of the independent sets of $G_{\bar{\theta}}$, for suitably chosen $\bar{\theta}$.

Theorem 1. *Let*

$$K^* = \big\{j \in [K] \colon p_{i^*,j} \geqslant 1 - O\big(\big(\tfrac{\log(KT)}{\Delta_j^2} + 1\big)^{-1/2}\big)\big\}$$

and for $\gamma \in [0, 1)$, let $\bar{\theta}(\gamma) = (\theta_1(\gamma), \ldots, \theta_K(\gamma))$ be defined as

$$\theta_i(\gamma) = \begin{cases} 1 - \gamma & \text{for } i \in K^* \\ 1 - \frac{O(\log(KT))}{\sqrt{T}} & \text{for } i \notin K^* \,. \end{cases}$$

The pseudo-regret R_T of UCB-DSG *over T rounds is bounded by:*

$$O\left(\min_{\gamma\in[0,1)}\left(\max_{I\in\mathcal{I}(G_{\bar{\theta}(\gamma)})}\sum_{j\in I}\frac{\ln(KT)\log(T)}{\Delta_j} + \frac{f(D,K)}{1-(1+\gamma)^2/4}\right)\right).$$

In the above, $f(D, K)$ is a function linear in the number of experts K and inversely proportional to each gap $\Delta \in D = \{\Delta_j : j \in [K], j \neq i^*\}$, but independent of T.

The denser the graph $G_{\bar{\theta}(\gamma)}$ the smaller the number of the independent sets, and thus the more favorable the bound is. Recalling that the independence number of a graph is the size of the maximum independent set of the graph, the number of terms in the above sum over any independence set, I, is upper bounded by the independence number, $\alpha(\gamma)$, of $G_{\bar{\theta}(\gamma)}$. Thus, in the case that $\Delta_j = \Delta$ for all $j \neq i^*$, this sum can be bounded by $\frac{\alpha(\gamma)}{\Delta}\ln(KT)\log(T)$. Moreover, $\gamma \in [0, 1)$ is a free parameter that allows for a limited tradeoff between the two terms in the sum. As γ increases, the number of independent sets decreases while the second term $f(D, K)/(1 - (1 + \gamma)^2/4)$ increases, and vice-versa.

Figure 1 contains an example of the thresholded process graph $G_{\bar{\theta}(\gamma)}$ defined in the theorem. Due to how the thresholds $\theta_i(\gamma)$ are defined, more edges connecting nodes within K^* are included in the graph $G_{\bar{\theta}(\gamma)}$, as compared to the others, thereby implying that the larger the set K^* is, the better the resulting regret bound. More concretely, if T is large and γ is a constant (say, $\gamma = 1/2$), then $1-\gamma \leqslant 1 - \frac{O(\log(KT))}{\sqrt{T}}$. Thus, edges connecting actions within K^* are more likely to be included in the thresholded graph $G_{\bar{\theta}(\gamma)}$ than edges crossing K^* or connecting nodes outside K^*, so that K^* is more likely to form a clique in $G_{\bar{\theta}(\gamma)}$. In this latter case, $\alpha(\gamma) \leqslant K - |K^*| + 1$ and hence, the bigger K^* the more favorable the regret bound is. Notice that the size of K^* is also influenced by the gaps Δ_js. In particular, for a given

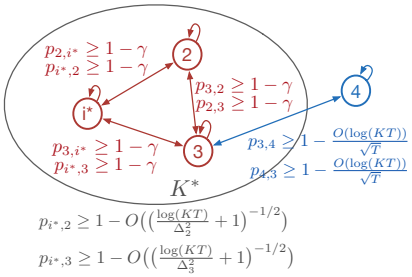

Figure 1. Example of the thresholded process graph $G_{\bar{\theta}(\gamma)}$. Experts ξ_{i^*}, ξ_2 and ξ_3 are in the set K^* as the probability of the incoming edge from i^* is big enough (bottom two inequalities in gray). Notice that $i^* \in K^*$ because of the self-loop. Edges between experts i and j in K^* are included in $G_{\bar{\theta}(\gamma)}$ if both $p_{i,j}$ and $p_{j,i}$ are $\geqslant 1-\gamma$. Moreover, since typically $1-\frac{O(\log(KT))}{\sqrt{T}} \geqslant 1-\gamma$, any edge connected to expert $\xi_4 \notin K^*$ must admit a probability greater than $1-\frac{O(\log(KT))}{\sqrt{T}}$ to be in the graph $G_{\bar{\theta}(\gamma)}$. Thus, more edges connecting experts within K^* are included in graph $G_{\bar{\theta}(\gamma)}$. The independence number of this graph is $\alpha(\gamma) = 2$.

matrix $[p_{i,j}]_{i,k=1}^K$, the smaller the gaps Δ_j (i.e., the harder the bandit problem), then the smaller K^*. For an expert ξ_j with a small gap Δ_j, the corresponding edge weight $p_{i^*,j}$ must be larger to be included in K^*. Said differently, if the best expert ξ_{i^*} is highly connected to experts with large expected losses, the pseudo-regret bound is favorable even if their connecting edge weights are comparatively small.

The standard UCB bound corresponds to a sum over all experts and since G admits self-loops at all vertices with weight one, our guarantee is always at least as favorable. In the special case where the probability matrix $[p_{i,j}]_{i,k=1}^K$ is binary and symmetric, then graphs G_t are deterministic, and all coincide with G. In this case, our bound coincides with that of Lykouris et al. (2020) [Theorem 3.2], which holds for a UCB-type algorithm, called UCB-N, that averages all available loss observations. In Appendix B (see Theorem 3 therein), we also prove a regret guarantee in terms of the clique partition number of $G_{\bar{\theta}(\gamma)}$ which, in the case of deterministic graphs, matches the clique-based bound for UCB-N in (Caron et al., 2012)[Theorem 2]. Even though summing over nodes of an independent set is smaller than summing over a clique cover, the bound based on cliques shaves off a $\log(T)$ factor compared to that in Theorem 1.

In the setting of fixed undirected graphs, Buccapatnam et al. (2014a) present an interesting action elimination algorithm that solves at every round a linear program based on the feedback graph. The authors prove favorable guarantees in terms of the minimum dominating set of the graph. Unfortunately, several difficulties arise when adapting (Buccapatnam et al., 2014a) to our setting of dependent stochastic graphs. From a computational standpoint, we would have to solve a linear program at every round based on the feedback graph's edge probabilities $\widehat{p}_{i,j}^{t-1}$. Moreover, we would need to know the time horizon T in advance (a key step of our analysis of UCB-DSG heavily depends on this property). One might instead estimate the $p_{i,j}$s in an initial stage where no actions are eliminated and then, say, adopt the strategy in (Li et al., 2020) that works in the case when the stochastic graph is *independent* of the losses and the $p_{i,j}$ are known. However, even this solution looks problematic in our setting. This is because the duration of this initial stage should be at least $\sqrt{T}$ in order to insure accurate estimation and, if we don't rely on feedback graphs during this stage we would suffer anyway a regret of the form $\frac{K}{\Delta}\log T$. More importantly, observe that even if we know the marginal probabilities $p_{i,j}$ *exactly*, we still have to face the loss bias issue intrinsic to our dependency assumptions.

4.1. Sketch of the Pseudo-regret Analysis

Here, we sketch the analysis that proves Theorem 1. Please see Appendix B for the details. At a high level, the proof proceeds via two steps: 1) we prove that $\widehat{\mu}_{i,t}$ converges to its mean at a rate of $\widetilde{O}(1/\sqrt{Q_{i,t}})$ and then bound the regret using a UCB-type analysis; 2) we relate the resulting bound to the independent sets of the process graph, $G_{\bar{\theta}(\gamma)}$.

For step 1), we first quantify the accuracy of our loss estimates $\widetilde{\ell}_t(\xi_j)$ (Lemma 1 and Lemma 2 in Appendix B). Specifically, Lemma 1 shows that even though $\widetilde{\nu}_{i,j,t}$ is a biased estimator, it still concentrates around $\mathbb{E}[\ell(\xi_i)\mathbb{I}\{i \notin N(j)\}]$. The key idea is to leverage the fact that the majority of observations used in this estimator are revealed from edges with large probability, thereby implying that this estimator mostly includes reliable observations. Then, Lemma 2 proves that conditioned on the past, $\widetilde{\ell}_t(\xi_i)$ converges to its expectation at a rate of $\widetilde{O}(1/\sqrt{U_{i,t}})$. By using the definition of threshold $\theta_{i,t-1}$ in conjunction with these two lemmas, Proposition 1 in Appendix B bounds the error term $\rho_{i,t}$ due to resorting to estimated losses when constructing estimator $\widehat{\mu}_{i,t}$:

$$\rho_{i,t} = \frac{1}{Q_{i,t}}\sum_{s=1}^{t}(\ell_s(\xi_i) - \widetilde{\ell}_s(\xi_i))\mathbb{I}\Big\{i \in \widehat{N}_s(I_s), i \notin N_s(I_s)\Big\}.$$

Proposition 1 states that $|\rho_{i,t}| \leqslant \widetilde{O}(1/\sqrt{Q_{i,t}})$. The interplay between the threshold and estimated loss is crucial in the proof of this proposition. In particular, it is this interplay that determines the threshold's key component $\sqrt{\frac{U_{i,t-1}}{Q_{i,t-1}}}$. This implies that $\widehat{\mu}_{i,t} + \rho_{i,t}$ is an unbiased estimate that concentrates around its mean μ_i and then, a UCB-type analysis shows that an expert $i \neq i^*$ cannot be pulled too often.

For step 2), we connect the sequence of graphs $\widehat{G}_1, \ldots, \widehat{G}_t$ generated by the algorithm to the thresholded process graph $G_{\bar{\theta}(\gamma)}$. Proposition 2 shows that the out-neighborhoods of $G_{\bar{\theta}(\gamma)}$ are contained in the out-neighborhoods of the graph $\widehat{G}_t$. In contrast to the thresholds $\theta_{i,t}$ generated by the al-

gorithm, the thresholds $\theta_i(\gamma)$ applied to process graph in Theorem 1 are both algorithm and time independent. To find such thresholds, we leverage a high probability statement on the number of times an expert is pulled (Lemma 3), and use several properties of $U_{i,t}$ and $Q_{i,t}$ including that $U_{i,t} \leqslant Q_{i,t} + 1$ (Lemma 4), thereby connecting such thresholds to the loss gaps, Δ_j. Lastly, following ideas in (Lykouris et al., 2020), we show in Proposition 3 that the regret bound depends on the the independent sets of graph $G_{\bar{\theta}(\gamma)}$.

4.2. Lower Bound

We now show that the intrinsic difficulty of learning with dependent stochastic feedback graphs does not derive from the stochasticity of the graphs (or the lack of prior knowledge of probabilities $p_{i,j}$) but, rather, from the arbitrariness of the dependency structure of graphs and losses.

Theorem 2. *For any number of arms $K \geqslant 2$, any gap value $\Delta \in (0, 1/4)$, any edge probability $p \in [0, 1)$, and any strongly consistent policy*[1] *there exists a dependent stochastic feedback graph problem with process graph G with $p_{i,j} = p$ for all $i \neq j$, $p_{i,i} = 1$ for all i, and expected losses $\mu_1, \ldots, \mu_K$, with $\mu_i - \mu_{i^*} = (1-p)\Delta$ for all $i \neq i^*$, such that $R_T = \Omega(\frac{K}{\Delta} \log T)$.*

In the above lower bound, p can be arbitrarily close to 1, but is assumed to be constant (independent of Δ, K, and T). Notice that this lower bound holds even if the algorithm knows p. Yet, this result does not violate the upper bound in Theorem 1, since that theorem refers to the independence number of an undirected graph $G_{\bar{\theta}(\gamma)}$ that retains enough edges (i, j) only when $|K^*|$ is big, which in turn happens only if the probabilities $p_{i^*,j}$ defining K^* increase to 1 as $T \to \infty$ and since $f(D, K)$ increases as the gaps decrease.

The lower bound also illustrates the stark difference between dependent and independent feedback graph problems. For instance, if G_t is an undirected Erdos-Renyi stochastic graph with probability p of independently generating each edge, then the gap-dependent pseudo-regret upper bound is of the form $\frac{\log K}{p\Delta} \log T$, i.e., logarithmic in K rather than linear. This conclusion can be extracted from both (Buccapatnam et al., 2014a) [Remark 5] and (Li et al., 2020) [Theorem 3], and essentially follows from the fact that the independence number of an Erdos-Renyi graph with K nodes and edge probability p is $\approx (2/p) \log K$ (Frieze, 1990).

5. Experiments

In this section, we present several experiments testing the UCB-DSG algorithm on multiclass classification problems. We focus on a regime where the number of rounds, $T = 10{,}000$, is relatively small compared to the number of experts, $K = 1{,}000$. This is a regime that fully illustrates the need to leverage the side information provided by feedback graphs, and where we expect vanilla UCB (one of our baselines) to experience slower convergence capabilities.

Consider the standard multi-class setting with c classes and the 0/1 loss $L(\xi(x), y) = \mathbb{I}\{\xi(x) \neq y\}$. Each expert, ξ_i for $i \in [K]$, consists of c hyperplanes, $[w_i^1, \ldots, w_i^c]$, each drawn randomly from a Gaussian $\mathcal{N}(0, 1)^d$, where d is the input dimension. Expert ξ_i labels input x according to the standard scoring function $\xi_i(x) = \text{argmax}_{y \in [c]} w_i^y \cdot x$. We define graphs G_t as follows: G_t admits a (bi-directional) edge from ξ_i to ξ_j if and only if $\xi_i(x_t) = \xi_j(x_t)$. If two experts predict the same class, then they admit the same loss and hence, their loss observability is precisely captured by the graph G_t. See Appendix A for further illustration.

We first compared UCB-DSG to the vanilla UCB algorithm, which only receives the loss of the chosen expert, and to the Fully Supervised (FS) algorithm, an unrealistic baseline violating our scenario, which at every round chooses the expert with the smallest empirical loss and receives the loss of every expert.

We present results for the `CIFAR` dataset, as well as ten datasets from the UCI repository: `letter`, `pendigits`, `poker`, `satimage`, `shuttle`, `segment`, `covtype`, `acoustic`, `HIGGS` and `sensorless` (see Appendix D for dataset statistics). The features of each of the UCI datasets were scaled to $[-1, 1]$. For the `CIFAR` dataset, we extracted via PCA the first 25 components and used them as features. The experiment was set up as follows. We randomly draw four times a set of hyperplanes, $\{[w_i^1, \ldots, w_i^c] : i \in [K]\}$, and for each set of hyperplanes, we randomly shuffle the data six times. Our results are averages over these 24 runs.

For all algorithms, since the constants in front of the log terms are artifacts of the analysis, we introduced a parameter $\lambda \in [0.1, 0.5, 1, 10, 100]$, and tuned each algorithm over this parameter as is standard practice. Specifically, the λ multiplies the slack terms in the confidence intervals. For UCB-DSG, we tune the algorithm over two λs: one for the slack, $S_{i,t}$, and threshold, $\theta_{i,t}$, as they both contain terms of the form $1/\sqrt{Q_{i,t}}$, and one for the estimated loss' conditions $C_{i,t}$, as it contains a term of the form $1/\sqrt{U_{i,t}}$. For each algorithm, we ran the above experiment for the different values of λ and report the results of the value of λ that admits the smallest regret at the last round.

Figure 2 (top row) shows the averaged regret results for some of the datasets and Appendix D contains the plots for all datasets. These figures show that UCB-DSG outperforms UCB on all eleven datasets except for two, for which it admits a comparable performance. UCB-DSG attains a performance that in a few cases is even close to FS.

[1] A strongly consistent policy is one such that $R_T = o(T^\alpha)$ for all $\alpha \in (0, 1)$.

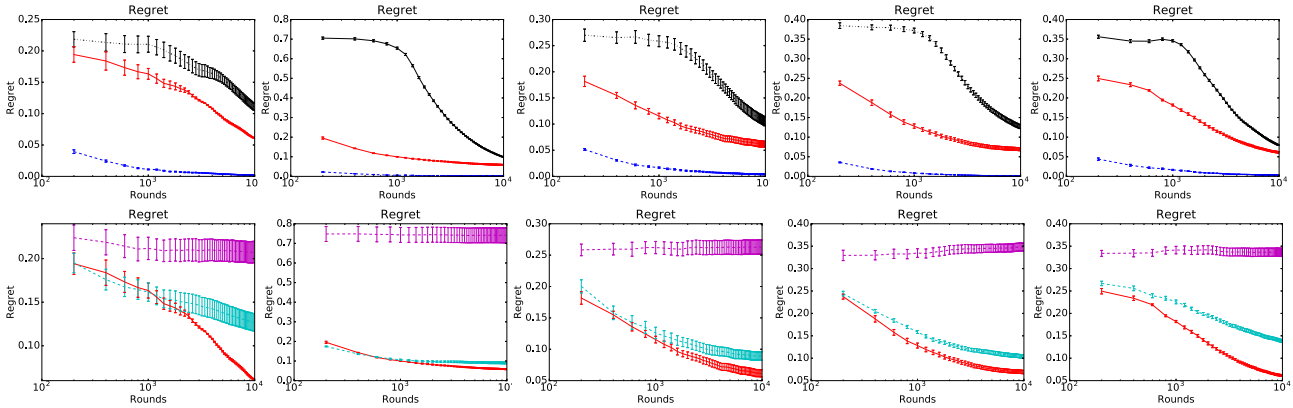

Figure 2. Top row is the average regret R_t/t as a function of t (log-scale) for UCB, UCB-DSG, and FS and the bottom row is the average regret R_t/t as a function of t (log-scale) for ALG-1, ALG-2, and UCB-DSG. Each column refers to different dataset ordered in the following way starting from the left: `pendigits`, `shuttle`, `segment`, `satimage`, and `covtype`.

We then compared UCB-DSG against two other baselines that we designed based on the concepts analyzed in this paper, ALG-1 and ALG-2. Both algorithms pick the expert with the smallest lower confidence bound $I_t = \operatorname{argmin}_{i\in[K]} \widehat{\mu}_{i,t-1} - S_{i,t-1}$, but the way they update the experts' empirical estimates differs.

The first baseline, ALG-1, is in the full information setting and is not relying on feedback graphs. ALG-1 updates the chosen expert by using its true loss, and uses $\ell'_t(\xi_i) = L(\xi_i(x_t), \xi_{I_t}(x_t))$ to update all other experts $i \neq I_t$. Note that, whenever ξ_{I_t} admits zero loss, all $\ell'_t(\xi_i) = \ell_t(\xi_i)$; however, ALG-1 could suffer a potentially linear regret, as the algorithm may add noisy labels at each round that corrupt the empirical estimates.

ALG-2 updates any expert that predicts the same way as ξ_{I_t} by using the true loss, that is, it updates all $i \in [K]$ that satisfy $\xi_i(x_t) = \xi_{I_t}(x_t)$ by using the loss $L(\xi_{I_t}(x_t), y_t)$. This algorithm is the natural extension of UCB-N of Caron et al. (2012) to this setting but, again, it may suffer linear regret since it relies on biased empirical estimates. To see why, suppose that ALG-2 selects ξ_1, and that $\xi_1(x) = \xi_2(x)$ only on those x such that $\xi_1(x) = \xi_2(x) = m$, for some $m \in [c]$. Then, the empirical estimate for ξ_2 may be biased, because it is averaged only over rounds t when $\xi_2(x_t) = m$.

Despite neither of these two algorithms are theoretically motivated, they are natural baselines. In our experiments (see the bottom row of Figure 2 as well as Appendix D), we found that UCB-DSG is in fact outperforming both baselines on all datasets except for three, `letter`, `HIGGS` and `CIFAR`. For `letter` and `CIFAR` datasets, UCB, ALG-1, ALG-2 and UCB-DSG all admit a similar performance, which seems to suggest that feedback graphs are not beneficial for these two datasets. For the `HIGGS` dataset, ALG-1, ALG-2 and UCB-DSG algorithms admit a performance close to FS, thereby indicating that given the current experimental setup, there is no opportunities for improvements for this dataset. In Appendix D, we further discuss how the density of the graphs affects the algorithms' performance.

Altogether, these experiments show that, in a natural scenario where feedback graph and losses are dependent, our algorithm UCB-DSG achieves a performance that is substantially more favorable than that of readily available baselines.

6. Conclusion

We initiated the analysis of online learning with stochastic feedback graphs in a setting often emerging in applications where graphs and losses are statistically dependent.

Our algorithm estimates the probability of an edge via past realizations of the stochastic graphs and, based on carefully chosen thresholds, uses estimated losses to update its empirical estimates. Our pseudo-regret bound is in terms of thresholded distributional properties of the process generating the feedback graphs. In this setting, our algorithm benefits from strong theoretical guarantees that become more favorable in cases where playing the best action tends to reveal more losses of the other actions. We have complemented our upper bound with a regret lower bound that illustrates the inherent hardness of the general dependent feedback graph setting. Finally, we report a series of experiments showing the favorable empirical performance of UCB-DSG, as compared to the UCB algorithm, as well as to readily available UCB-based variants of UCB-DSG that propagate information across actions in a way not supported by our theory.

It can be shown that, under some assumptions, our analysis can be extended to the more general setting where feedback graphs may not admit self-loops at all vertices. This is an important setting that includes naturally *active learning* scenarios, as discussed in Appendix C.

References

Alon, N., Cesa-Bianchi, N., Gentile, C., and Mansour, Y. From bandits to experts: A tale of domination and independence. In *NIPS*, 2013.

Alon, N., Cesa-Bianchi, N., Dekel, O., and Koren, T. Online learning with feedback graphs: Beyond bandits. In *JMLR*, pp. 23–35, 2015.

Alon, N., Cesa-Bianchi, N., Gentile, C., Mannor, S., Mansour, Y., and Shamir, O. Nonstochastic multi-armed bandits with graph-structured feedback. *SIAM J. Comput.*, 46(6):1785–1826, 2017.

Audibert, J. Y., Munos, R., and Szepesvári, C. Exploration-exploitation trade-off using variance estimates in multi-armed bandits. In *Theoretical Computer Science*, 2009.

Auer, P., Cesa-Bianchi, N., and Fischer, P. Finite-time analysis of the multiarmed bandit problem. *Mach. Learn.*, 47(2-3):235–256, 2002a.

Auer, P., Cesa-Bianchi, N., and Gentile, C. Adaptive and self-confident on-line learning algorithms. *Journal of Computer and System Sciences*, 64, 2002b.

Buccapatnam, S., Eryilmaz, A., and Shroff, N. Stochastic bandits with side observations on networks. In *SIGMETRICS*, 2014a.

Buccapatnam, S., Eryilmaz, A., and Shroff, N. B. Stochastic bandits with side observations on networks. In *The 2014 ACM International Conference on Measurement and Modeling of Computer Systems*, SIGMETRICS '14, pp. 289–300. ACM, 2014b.

Burnetas, A. and Katehakis, M. Optimal adaptive policies for sequential allocation problems. *Advances in Applied Mathematics*, 17(2):122–142, 1996.

Caron, S., Kveton, B., Lelarge, M., and Bhagat, S. Leveraging side observations in stochastic bandits. In *UAI*, 2012.

Cesa-Bianchi, N. and Lugosi, G. *Prediction, Learning, and Games*. Cambridge University Press, New York, NY, USA, 2006.

Cohen, A., Hazan, T., and Koren, T. Online learning with feedback graphs without the graphs. *ICML*, 2016.

Cortes, C., DeSalvo, G., and Mohri, M. Learning with rejection. In *ALT*, 2016.

Cortes, C., DeSalvo, G., Gentile, C., Mohri, M., and Yang, S. Online learning with abstention. In *35th ICML*, 2018.

Cortes, C., DeSalvo, G., Mohri, M., Gentile, C., and Yang, S. Online learning with sleeping experts and feedback graphs. In *ICML*, 2019.

Frieze, A. M. On the independence number of random graphs. *Discrete Mathematics*, 81:171–175, 1990.

Kocák, T., Neu, G., Valko, M., and Munos, R. Efficient learning by implicit exploration in bandit problems with side observations. In *NIPS*, pp. 613–621, 2014.

Kocák, T., Neu, G., and Valko, M. Online learning with noisy side observations. *AISTATS*, 2016.

Lai, T. and Robbins, H. Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics*, 6(1): 4–22, 1985.

Li, S., Chen, W., and Leung, K. Stochastic online learning with probabilistic graph feedback. In *34st AAAI Conference on Artificial Intelligence*, 2020.

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

Liu, F., Buccapatnam, S., and Shroff, N. Information directed sampling for stochastic bandits with graph feedback. In *32nd AAAI Conference on Artificial Intelligence*, 2018.

Lykouris, T., Tardos, E., and Wali, D. Feedback graphs regret bounds for thompson sampling and ucb. *ALT*, 2020.

Mannor, S. and Shamir, O. From bandits to experts: On the value of side-observations. *NIPS*, pp. 291–307, 2011.

Neu, G. Explore no more: Improved high-probability regret bounds for non-stochastic bandits. In *NIPS*, pp. 3168–3176, 2015.

Salomon, A., Audibert, J., and El Alaoui, I. Lower bounds and selectivity of weak-consistent policies in stochastic multi-armed bandit problem. *Journal of Machine Learning Research*, 14:187–207, 2013.

Tossou, A., Dimitrakakis, C., and Dubhashi, D. Thompson sampling for stochastic bandits with graph feedback. In *31st AAAI Conference on Artificial Intelligence*, 2017.

van de Geer, S. On hoeffding's inequality for dependent random variables. *Dehling H., Mikosch T., Sørensen M. (eds) Empirical Process Techniques for Dependent Data*, 2002.

Wu, Y., György, A., and Szepesvari, C. Online learning with Gaussian payoffs and side observations. In *Advances in Neural Information Processing Systems 28*, pp. 1360–1368. Curran Associates, Inc., 2015a.

Wu, Y., György, A., and Szepesvári, C. Online learning with Gaussian payoffs and side observations. In *NIPS*, pp. 1360–1368, 2015b.

Yun, D., Proutiere, A., Ahn, S., Shin, J., and Yi, Y. Multi-armed bandit with additional observations. *Proc. ACM Meas. Anal. Comput. Syst.*, 2(1):13:1–13:22, 2018.