Online Learning under Delayed Feedback

Pooria Joulani POORIA@UALBERTA.CA
András György GYORGY@UALBERTA.CA
Csaba Szepesvári SZEPESVA@UALBERTA.CA
Dept. of Computing Science, University of Alberta, Edmonton, AB, T6G 2E8 CANADA

Abstract

Online learning with delayed feedback has received increasing attention recently due to its several applications in distributed, web-based learning problems. In this paper we provide a systematic study of the topic, and analyze the effect of delay on the regret of online learning algorithms. Somewhat surprisingly, it turns out that delay increases the regret in a multiplicative way in adversarial problems, and in an additive way in stochastic problems. We give meta-algorithms that transform, in a black-box fashion, algorithms developed for the non-delayed case into ones that can handle the presence of delays in the feedback loop. Modifications of the well-known UCB algorithm are also developed for the bandit problem with delayed feedback, with the advantage over the meta-algorithms that they can be implemented with lower complexity.

1. Introduction

In this paper we study sequential learning when the feedback about the predictions made by the forecaster are delayed. This is the case, for example, in web advertisement, where the information whether a user has clicked on a certain ad may come back to the engine in a delayed fashion: after an ad is selected, while waiting for the information if the user clicks or not, the engine has to provide ads to other users. Also, the click information may be aggregated and then periodically sent to the module that decides about the ads, resulting in further delays. (Li et al., 2010; Dudik et al., 2011). Another example is parallel, distributed learning, where propagating information among nodes causes delays (Agarwal & Duchi, 2011).

Proceedings of the 30^{th} International Conference on Machine Learning, Atlanta, Georgia, USA, 2013. JMLR: W&CP volume 28.

While online learning has proved to be successful in many machine learning problems and is applied in practice in situations where the feedback is delayed, the theoretical results for the non-delayed setup are not applicable when delays are present. Previous work concerning the delayed setting focussed on specific online learning settings and delay models (mostly with constant delays). Thus, a comprehensive understanding of the effects of delays is missing. In this paper, we provide a systematic study of online learning problems with delayed feedback. We consider the *partial monitoring setting*, which covers all settings previously considered in the literature, extending, unifying, and often improving upon existing results. In particular, we give general meta-algorithms that transform, in a black-box fashion, algorithms developed for the non-delayed case into algorithms that can handle delays efficiently. We analyze how the delay effects the regret of the algorithms. One interesting, perhaps somewhat surprising, result is that the delay inflates the regret in a multiplicative way in adversarial problems, while this effect is only additive in stochastic problems. While our general meta-algorithms are useful, their time- and space-complexity may be unnecessarily large. To resolve this problem, we work out modifications of variants of the UCB algorithm (Auer et al., 2002) for stochastic bandit problems with delayed feedback that have much smaller complexity than the black-box algorithms.

The rest of the paper is organized as follows. The problem of online learning with delayed feedback is defined in Section 2. The adversarial and stochastic problems are analyzed in Sections 3.1 and 3.2, while the modification of the UCB algorithm is given in Section 4. Due to space limitations, some proofs are omitted and are included only in the extended version of this paper (Joulani et al., 2013).

2. The delayed feedback model

We consider a general model of online learning, which we call the partial monitoring problem with side in-

Parameters: Forecaster's prediction set $\mathcal{A}$, set of outcomes $\mathcal{B}$, side information set $\mathcal{X}$, reward function $r : \mathcal{X} \times \mathcal{A} \times \mathcal{B} \to \mathbb{R}$, feedback function $h : \mathcal{X} \times \mathcal{A} \times \mathcal{B} \to \mathcal{H}$, time horizon n (optional).
At each time instant $t = 1, 2, \ldots, n$:

1. The environment chooses some side information $x_t \in \mathcal{X}$ and an outcome $b_t \in \mathcal{B}$.
2. The side information x_t is presented to the forecaster, who makes a prediction $a_t \in \mathcal{A}$, which results in the reward $r(x_t, a_t, b_t)$ (unknown to the forecaster).
3. The feedback $h_t = h(x_t, a_t, b_t)$ is scheduled to be revealed after τ_t time instants.
4. The agent observes $H_t = \{(t', h_{t'}) : t' \le t, t' + \tau_{t'} = t\}$, i.e., all the feedback values scheduled to be revealed at time step t, together with their timestamps.

Figure 1: Partial monitoring under delayed, time-stamped feedback.

formation. In this model, the forecaster (decision maker) has to make a sequence of predictions (actions), possibly based on some side information, and for each prediction it receives some reward and feedback, where the feedback is delayed. More formally, given a set of possible side information values $\mathcal{X}$, a set of possible predictions $\mathcal{A}$, a set of reward functions $\mathcal{R} \subset \{r : \mathcal{X} \times \mathcal{A} \to \mathbb{R}\}$, and a set of possible feedback values $\mathcal{H}$, at each time instant $t = 1, 2, \ldots$, the forecaster receives some side information $x_t \in \mathcal{X}$; then, possibly based on the side information, the forecaster predicts some value $a_t \in \mathcal{A}$ while the environment simultaneously chooses a reward function $r_t \in \mathcal{R}$; finally, the forecaster receives reward $r_t(x_t, a_t)$ and some time-stamped feedback set $H_t \subset \mathbb{N} \times \mathcal{H}$. In particular, each element of H_t is a pair of time index and a feedback value, the time index indicating the time instant whose decision the associated feedback corresponds to.

Note that the forecaster may or may not receive any direct information about the rewards it receives (i.e., the rewards may be hidden). In standard online learning, the feedback-set H_t is a singleton and the feedback in this set depends on r_t, a_t. In the delayed model, however, the feedback that concerns the decision at time t is received at the end of the time period $t+\tau_t$, *after* the prediction is made, i.e., it is delayed by τ_t time steps. Note that $\tau_t \equiv 0$ corresponds to the non-delayed case. Due to the delays multiple feedbacks may arrive at the same time, hence the definition of H_t.

The goal of the forecaster is to maximize its cumulative reward $\sum_{t=1}^{n} r_t(x_t, a_t)$ $(n \ge 1)$. The performance of the forecaster is measured relative to the best static strategy selected from some set $\mathcal{F} \subset \{f \,|\, f : \mathcal{X} \to \mathcal{A}\}$ in hindsight. In particular, the forecaster's performance is measured through the *regret*, defined by

$$R_n = \sup_{a \in \mathcal{F}} \sum_{t=1}^{n} r_t(x_t, a(x_t)) - \sum_{t=1}^{n} r_t(x_t, a_t).$$

A forecaster is consistent if it achieves, asymptotically, the average reward of the best static strategy, that is $\mathbb{E}[R_n]/n \to 0$, and we are interested in how fast the average regret can be made to converge to 0.

The above general problem formulation includes most scenarios considered in online learning. In the full information case, the feedback is the reward function itself, that is, $\mathcal{H} = \mathcal{R}$ and $H_t = \{(t, r_t)\}$ (in the non-delayed case). In the bandit case, the forecaster only learns the rewards of its own prediction, i.e., $\mathcal{H} = \mathbb{R}$ and $H_t = \{(t, r_t(x_t, a_t))\}$. In the partial monitoring case, the forecaster is given a reward function $r : \mathcal{X} \times \mathcal{A} \times \mathcal{B} \to \mathbb{R}$ and a feedback function $h : \mathcal{X} \times \mathcal{A} \times \mathcal{B} \to \mathcal{H}$, where $\mathcal{B}$ is a set of choices (outcomes) of the environment. Then, for each time instant the environment picks an outcome $b_t \in \mathcal{B}$, and the reward becomes $r_t(x_t, a_t) = r(x_t, a_t, b_t)$, while $H_t = \{(t, h(x_t, a_t, b_t))\}$. This interaction protocol is shown in Figure 1 in the delayed case. Note that the bandit and full information problems can also be treated as special partial monitoring problems. Therefore, we will use this last formulation of the problem. When no stochastic assumption is made on how the sequence b_t is generated, we talk about the adversarial model. In the stochastic setting we will consider the case when b_t is a sequence of independent, identically distributed (i.i.d.) random variables. Side information may or may not be present in a real problem; in its absence $\mathcal{X}$ is a singleton set.

Finally, we may have different assumptions on the delays. Most often, we will assume that $(\tau_t)_{t \ge 1}$ is an i.i.d. sequence, which is independent of the past predictions $(a_s)_{s \le t}$ of the forecaster. In the stochastic setting, we also allow the distribution of τ_t to depend on a_t.

Note that the delays may change the order of observing the feedbacks, with the feedback of a more recent prediction being observed before the feedback of an earlier one.

2.1. Related work

The effect of delayed feedback has been studied in the recent years under different online learning scenarios

		Stochastic Feedback	General (Adversarial) Feedback
Full Info	No Side Info	$R(n) \leq R'(n) + O(\mathbb{E}\left[\tau_t^2\right])$ (Agarwal & Duchi, 2011)	L $R(n) \leq O(\tau_{const}) \times R'(n/\tau_{const})$ (Weinberger & Ordentlich, 2002) (Langford et al., 2009) (Agarwal & Duchi, 2011)
	Side Info	L $R(n) \leq R'(n) + O(D^*)$ (Mesterharm, 2007)	L $R(n) \leq O(\bar{D}) \times R'(n/\bar{D})$ (Mesterharm, 2007)
Bandit Feedback	No Side Info	$R(n) \leq C_1 R'(n) + C_2 \tau_{\max} \log(\tau_{\max})$ (Desautels et al., 2012)	$R(n) \leq O(\tau_{const}) \times R(n/\tau_{const})$ (Neu et al., 2010)
	Side Info	$R(n) \leq R'(n) + O(\tau_{const}\sqrt{\log n})$ (Dudik et al., 2011)	
Partial Monitoring	No Side Info	$\mathbf{R_n \leq R'(n) + O(G_n^*)}$	$\mathbf{R_n \leq (1 + \mathbb{E}\left[G_n^*\right]) \times R'\left(\frac{n}{1 + \mathbb{E}\left[G_n^*\right]}\right)}$
	Side Info		$\mathbf{R_n \leq (1 + \mathbb{E}\left[G_n^*\right]) \times R'\left(\frac{n}{1 + \mathbb{E}\left[G_n^*\right]}\right)}$

Table 1. Summary of work on online learning under delayed feedback. $R(n)$ shows the (expected) regret in the delayed setting, while $R'(n)$ shows the (upper bound on) the (expected) regret in the non-delayed setting. L denotes a matching lower bound. D^* and $\bar{D}$ indicate the maximum and average *gap*, respectively, where a gap is a number of consecutive time steps the agent does not get any feedback (in the adversarial delay formulation used by Mesterharm (2005; 2007)). The term τ_{const} indicates that the results are for constant delays only. For the work of (Desautels et al., 2012), C_1 and C_2 are positive constants, with $C_1 > 1$, and $\tau_{\max}$ denotes the maximum delay. The results presented in this paper are shown in boldface, where $\mathbf{G_t^*}$ is the maximum number of outstanding feedbacks during the first t time-steps. In particular, $\mathbf{G_n^* \leq \tau_{max}}$ when the delays have an upper bound τ_{max}, and we show that $\mathbf{G_n^* = O\left(\mathbb{E}\left[\tau_t\right] + \sqrt{\mathbb{E}\left[\tau_t\right] \log n} + \log n\right)}$ when the delays τ_t are i.i.d. The new bounds for the partial monitoring problem are automatically applicable in the other, spacial, cases, and give improved results in most cases.

and different assumptions on the delay. A concise summary, together with the contributions of this paper, is given in Table 1.

To the best of our knowledge, Weinberger & Ordentlich (2002) were the first to analyze the delayed feedback problem; they considered the adversarial full information setting with a fixed, known delay τ_{const}. They showed that the minimax optimal solution is to run $\tau_{const} + 1$ independent optimal predictors on the subsampled reward sequences: $\tau_{const} + 1$ prediction strategies are used such that the i^{th} predictor is used at time instants t with $(t \mod (\tau_{const} + 1)) + 1 = i$. This approach forms the basis of our method devised for the adversarial case (see Section 3.1). Langford et al. (2009) showed that under the usual conditions, a sufficiently slowed-down version of the mirror descent algorithm achieves optimal decay rate of the average regret. Mesterharm (2005; 2007) considered another variant of the full information setting, using an adversarial model on the delays in the label prediction setting, where the forecaster has to predict the label corresponding to a side information vector x_t. While in the full information online prediction problem Weinberger & Ordentlich (2002) showed that the regret increases by a multiplicative factor of τ_{const}, in the work of Mesterharm (2005; 2007) the important quantity becomes the maximum/average gap defined as the length of the largest time interval the forecaster does not receive feedback. Mesterharm (2005; 2007) also shows that the minimax regret in the adversarial case increases multiplicatively by the average gap, while it increases only in an additive fashion in the stochastic case, by the maximum gap. Agarwal & Duchi (2011) considered the problem of online stochastic optimization and showed that, for i.i.d. random delays, the regret increases with an additive factor of order $\mathbb{E}\left[\tau^2\right]$.

Qualitatively similar results were obtained in the bandit setting. Considering a fixed and known delay τ_{const}, Dudik et al. (2011) showed an additive $O(\tau_{const}\sqrt{\log n})$ penalty in the regret for the stochastic setting (with side information), while (Neu et al., 2010) showed a multiplicative regret for the adversarial bandit case. The problem of delayed feedback has also been studied for Gaussian process bandit optimization (Desautels et al., 2012), resulting in a multiplicative

increase in the regret that is independent of the delay and an additive term depending on the maximum delay.

In the rest of the paper we generalize the above results to the partial monitoring setting, extending, unifying, and often improving existing results.

3. Black-Box Algorithms for Delayed Feedback

In this section we provide black-box algorithms for the delayed feedback problem. We assume that there exists a base algorithm BASE for solving the prediction problem without delay. We often do not specify the assumptions underlying the regret bounds of these algorithms, and assume that the problem we consider only differs from the original problem because of the delays. For example, in the adversarial setting, BASE may build on the assumption that the reward functions are selected in an oblivious or non-oblivious way (i.e., independently or not of the predictions of the forecaster). First we consider the adversarial case in Section 3.1. Then in Section 3.2, we provide tighter bounds for the stochastic case.

3.1. Adversarial setting

We say that a prediction algorithm *enjoys a regret or expected regret bound* $f : [0,\infty) \to \mathbb{R}$ under the given assumptions in the non-delayed setting if (i) f is nondecreasing, concave, $f(0) = 0$; and (ii) $\sup_{b_1,\dots,b_n \in \mathcal{B}} R_n \le f(n)$ or, respectively, $\sup_{b_1,\dots,b_n \in \mathcal{B}} \mathbb{E}[R_n] \le f(n)$ for all n. The algorithm of Weinberger & Ordentlich (2002) for the adversarial full information setting subsamples the reward sequence by the constant delay $\tau_{const}+1$, and runs a base algorithm BASE on each of the $\tau_{const}+1$ subsampled sequences. Weinberger & Ordentlich (2002) showed that if BASE enjoys a regret bound f then their algorithm in the fixed delay case enjoys a regret bound $(\tau_{const}+1)f(n/(\tau_{const}+1))$. Furthermore, when BASE is minimax optimal in the non-delayed setting, the subsampling algorithm is also minimax optimal in the (full information) delayed setting, as can be seen by constructing a reward sequence that changes only in every $\tau_{const}+1$ times. Note that Weinberger & Ordentlich (2002) do not require condition (i) of f. However, these conditions imply that $yf(x/y)$ is a concave function of y for any fixed x (a fact which will turn out to be useful in the analysis later), and are satisfied by all regret bounds we are aware of (e.g., for multi-armed bandits, contextual bandits, partial monitoring, etc.), which all have a regret upper bound of the form $\widetilde{O}(n^\alpha)$ for some $0 \le \alpha \le 1$, with, typically, $\alpha = 1/2$ or $2/3$.[1].

In this section we extend the algorithm of Weinberger & Ordentlich (2002) to the case when the delays are not constant, and to the partial monitoring setting. The idea is that we run several instances of a non-delayed algorithm BASE as needed: an instance is "free" if it has received the feedback corresponding to its previous prediction – before this we say that the instance is "busy", waiting for the feedback. When we need to make a prediction, we use one of existing instances that is free, and is hence ready to make another prediction. If no such instance exists, we create a new one to be used (a new instance is always "free", as it is not waiting for the feedback of a previous prediction). The resulting algorithm, which we call Black-Box Online Learning under Delayed feedback (BOLD) is shown below (note that when the delays are constant, BOLD reduces to the algorithm of Weinberger & Ordentlich (2002)):

Algorithm 1 Black-box Online Learning under Delayed feedback (**BOLD**)

for each time instant $t = 1, 2, \dots, n$ **do**
  Prediction:
    Pick a free instance of BASE (independently of past predictions), or create a new instance if all existing instances are busy. Feed the instance picked with x_t and use its prediction.
  Update:
  for each $(s, h_s) \in H_t$ **do**
    Update the instance used at time instant s with the feedback h_s.
  end for
end for

Clearly, the performance of BOLD depends on how many instances of BASE we need to create, and how many times each instance is used. Let M_t denote the number of BASE instances created by BOLD up to and including time t. That is, $M_1 = 1$, and we create a new instance at the beginning of any time instant when all instances are waiting for their feedback. Let $G_t = \sum_{s=1}^{t-1} \mathbb{I}\{s + \tau_s \ge t\}$ be the total number of outstanding (missing) feedbacks when the forecaster is making a prediction at time instant t. Then we have G_t algorithms waiting for their feedback, and so $M_t \ge G_t + 1$. Since we only introduce new instances when it is necessary (and each time instant at most one new instance is created), it is easy to see that

$$M_t = G_t^* + 1 \tag{1}$$

[1] $u_n = \widetilde{O}(v_n)$ means that there is a $\beta \ge 0$ such that $\lim_{n\to\infty} u_n/(v_n \log^\beta n) = 0$.

for any t, where $G_t^* = \max_{1 \le s \le t} G_t$.

We can use the result above to transfer the regret guarantee of the non-delayed base algorithm BASE to a guarantee on the regret of BOLD.

Theorem 1. *Suppose that the non-delayed algorithm* BASE *used in BOLD enjoys an (expected) regret bound f_{BASE}. Assume, furthermore, that the delays τ_t are independent of the forecaster's prediction a_t. Then the expected regret of BOLD after n time steps satisfies*

$$\begin{aligned}\mathbb{E}[R_n] &\le \mathbb{E}\left[(G_n^*+1) f_{\text{BASE}}\left(\frac{n}{G_n^*+1}\right)\right] \\ &\le (\mathbb{E}[G_n^*]+1) f_{\text{BASE}}\left(\frac{n}{\mathbb{E}[G_n^*]+1}\right).\end{aligned}$$

Proof. As the second inequality follows from the concavity of $y \mapsto y f_{\text{BASE}}(x/y)$ $(x, y > 0)$, it remains to prove the first one.

For any $1 \le j \le M_n$, let L_j denote the list of time instants in which BOLD has used the prediction chosen by instance j, and let $n_j = |L_j|$ be the number of time instants this happens. Furthermore, let $R_{n_j}^j$ denote the regret incurred during the time instants t with $t \in L_j$:

$$R_{n_j}^j = \sup_{a \in \mathcal{F}} \sum_{t \in L_j} r_t(x_t, a(x_t)) - \sum_{t \in L_j} r_t(x_t, a_t),$$

where a_t is the prediction made by BOLD (and instance j) at time instant t. By construction, instance j does not experience any delays. Hence, $R_{n_j}^j$ is its regret in a non-delayed online learning problem. [2] Then,

$$\begin{aligned}R_n &= \sup_{a \in \mathcal{F}} \sum_{t=1}^{n} r_t(x_t, a(x_t)) - \sum_{t=1}^{n} r_t(x_t, a_t) \\ &= \sup_{a \in \mathcal{F}} \sum_{j=1}^{M_n} \sum_{t \in L_j} r_t(x_t, a(x_t)) - \sum_{j=1}^{M_n} \sum_{t \in L_j} r_t(x_t, a_t) \\ &\le \sum_{j=1}^{M_n} \left(\sup_{a \in \mathcal{F}} \sum_{t \in L_j} r_t(x_t, a(x_t)) - \sum_{t \in L_j} r_t(x_t, a_t) \right) \\ &= \sum_{j=1}^{M_n} R_{n_j}^j.\end{aligned}$$

Now, using the fact that f_{BASE} is an (expected) regret bound, we obtain

$$\begin{aligned}\mathbb{E}[R_n | \tau_1, \dots, \tau_n] &\le \sum_{j=1}^{M_n} \mathbb{E}\left[R_{n_j}^j | \tau_1, \dots, \tau_n\right] \\ &\le \sum_{j=1}^{M_n} f_{\text{BASE}}(n_j) = M_n \sum_{j=1}^{M_n} \frac{1}{M_n} f_{\text{BASE}}(n_j) \\ &\le M_n f_{\text{BASE}}\left(\sum_{j=1}^{M_n} \frac{1}{M_n} n_j\right) = M_n f_{\text{BASE}}\left(\frac{n}{M_n}\right),\end{aligned}$$

where the first inequality follows since M_n is a deterministic function of the delays, while the last inequality follows from Jensen's inequality and the concavity of f_{BASE}. Substituting M_n from (1) and taking the expectation concludes the proof. □

Now, we need to bound G_n^* to make the theorem meaningful. When all delays are the same constants, for $n > \tau_{const}$ we get $G_n^* = \tau_t = \tau_{const}$, and we get back the regret bound

$$\mathbb{E}[R_n] \le (\tau_{const}+1) f_{\text{BASE}}\left(\frac{n}{\tau_{const}+1}\right)$$

of Weinberger & Ordentlich (2002), thus generalizing their result to partial monitoring. We do not know whether this bound is tight even when BASE is minimax optimal, as the argument of Weinberger & Ordentlich (2002) for the lower bound does not work in the partial information setting (the forecaster can gain extra information in each block with the same reward functions).

Assuming the delays are i.i.d., we can give an interesting bound on G_n^*. The result is based on the fact that although G_t can be as large as t, both its expectation and variance are upper bounded by $\mathbb{E}[\tau_1]$.

Lemma 2. *Assume $\tau_1, \dots, \tau_n$ is a sequence of i.i.d. random variables with finite expected value, and let $B(n,t) = t + 2\log n + \sqrt{4t \log n}$. Then*

$$\mathbb{E}[G_n^*] \le B(n, \mathbb{E}[\tau_1]) + 1.$$

Proof. First consider the expectation and the variance of G_t. For any t,

$$\begin{aligned}\mathbb{E}[G_t] &= \mathbb{E}\left[\sum_{s=1}^{t-1} \mathbb{I}\{s + \tau_s \ge t\}\right] = \sum_{s=1}^{t-1} \mathbb{P}\{s + \tau_s \ge t\} \\ &= \sum_{s=0}^{t-2} \mathbb{P}\{\tau_1 > s\} \le \mathbb{E}[\tau_1],\end{aligned}$$

[2] Note that L_j is a function of the delay sequence and is not a function of the predictions $(a_t)_{t \ge 1}$. Hence, the reward sequence that instance j is evaluated on is chosen obliviously whenever the adversary of BOLD is oblivious.

and, similarly

$$\sigma^2[G_t] = \sum_{s=1}^{t-1} \sigma^2\left[\mathbb{I}\{s+\tau_s \geq t\}\right] \leq \sum_{s=1}^{t-1} \mathbb{P}\{s+\tau_s \geq t\},$$

so $\sigma^2[G_t] \leq \mathbb{E}[\tau_1]$ in the same way as above. By Bernstein's inequality (Cesa-Bianchi & Lugosi, 2006, Corollary A.3), for any $0 < \delta < 1$ and any t we have, with probability at least $1-\delta$,

$$G_t - \mathbb{E}[G_t] \leq \log\tfrac{1}{\delta} + \sqrt{2\sigma^2[G_t]\log\tfrac{1}{\delta}}.$$

Applying the union bound for $\delta = 1/n^2$, and our previous bounds on the variance and expectation of G_t, we obtain that with probability at least $1-1/n$,

$$\max_{1\leq t\leq n} G_t \leq \mathbb{E}[\tau_1] + 2\log n + \sqrt{4\mathbb{E}[\tau_1]\log n}.$$

Taking into account that $\max_{1\leq t\leq n} G_t \leq n$, we get the statement of the lemma. □

Corollary 3. *Under the conditions of Theorem 1, if the sequence of delays is i.i.d, then*

$$\mathbb{E}[R_n] \leq (B(n,\mathbb{E}[\tau_1])+2) f_{\textsc{Base}}\left(\frac{n}{B(n,\mathbb{E}[\tau_1])+2}\right).$$

Note that although the delays can be arbitrarily large, whenever the expected value is finite, the bound only increases by a $\log n$ factor.

3.2. Finite stochastic setting

In this section, we consider the case when the prediction set $\mathcal{A}$ of the forecaster is finite; without loss of generality we assume $\mathcal{A} = \{1,2,\ldots,K\}$. We also assume that there is no side information (that is, x_t is a constant for all t, and, hence, will be omitted; the results can be extended easily to the case of a finite side information set, where we can repeat the procedures described below for each value of the side information separately). The main assumption in this section is that the outcomes $(b_t)_{t\geq 1}$ form an i.i.d. sequence, which is also independent of the predictions of the forecaster. When $\mathcal{B}$ is finite, this leads to the standard i.i.d. partial monitoring (IPM) setting, while the conventional multi-armed bandit (MAB) setting is recovered when the feedback is the reward of the last prediction, that is, $h_t = r_t(a_t, b_t)$. As in the previous section, we will assume that the feedback delays are independent of the outcomes of the environment. The main result of this section shows that under these assumptions, the penalty in the regret grows in an additive fashion due to the delays, as opposed to the multiplicative penalty that we have seen in the adversarial case.

By the independence assumption on the outcomes, the sequences of potential rewards $r_t(i) \doteq r(i,b_t)$ and feedbacks $h_t(i) \doteq h(i,b_t)$ are i.i.d., respectively, for the same prediction $i \in \mathcal{A}$. In this setting we also assume that the feedback and reward sequences of different predictions are independent of each other. Let $\mu_i = \mathbb{E}[r_t(i)]$ denote the expected reward of predicting i, $\mu^* = \max_{i\in\mathcal{A}} \mu_i$ the optimal reward and i^* with $\mu_{i^*} = \mu^*$ the optimal prediction. Moreover, let $T_i(n) = \sum_{t=1}^{n} \mathbb{I}\{a_t = i\}$ denote the number of times i is predicted by the end of time instant n. Then, defining the "gaps" $\Delta_i = \mu^* - \mu_i$ for all $i \in \mathcal{A}$, the expected regret of the forecaster becomes

$$\mathbb{E}[R_n] = \sum_{t=1}^{n} \mu^* - \mu_{a_t} = \sum_{i=1}^{K} \Delta_i \mathbb{E}[T_i(n)]. \qquad (2)$$

Similarly to the adversarial setting, we build on a base algorithm Base for the non-delayed case. The advantage in the IPM setting (and that we consider expected regret) is that here Base can consider a permuted order of rewards and feedbacks, and so we do not have to wait for the actual feedback; it is enough to receive a feedback for the same prediction. This is the idea at the core of our algorithm, Queued Partial Monitoring with Delayed Feedback (QPM-D):

Algorithm 2 Queued Partial Monitoring with Delays (QPM-D)

```
Create an empty FIFO buffer Q[i] for each i ∈ A.
Let I be the first prediction of Base.
for each time instant t = 1, 2, ..., n do
  Predict:
  while Q[I] is not empty do
    Update Base with a feedback from Q[I].
    Let I be the next prediction of Base.
  end while
  There are no buffered feedbacks for I, so predict
  a_t = I at time instant t to get a feedback.
  Update:
  for each (s, h_s) ∈ H_t do
    Add the feedback h_s to the buffer Q[a_s].
  end for
end for
```

Here we have a Base partial monitoring algorithm for the non-delayed case, which is run inside the algorithm. The feedback information coming from the environment is stored in separate queues for each prediction value. The outer algorithm constantly queries Base: while feedbacks for the predictions made are available in the queues, only the inner algorithm Base runs (that is, this happens within a single time instant

in the real prediction problem). When no feedback is available, the outer algorithm keeps sending the same prediction to the real environment until a feedback for that prediction arrives. In this way BASE is run in a simulated non-delayed environment. The next lemma implies that the inner algorithm BASE actually runs in a non-delayed version of the problem, as it experiences the same distributions:

Lemma 4. *Consider a delayed stochastic IPM problem as defined above. For any prediction i, for any $s \in \mathbb{N}$ let $h'_{i,s}$ denote the s^{th} feedback QPM-D receives for predicting i. Then the sequence $(h'_{i,s})_{s\in\mathbb{N}}$ is an i.i.d. sequence with the same distribution as the sequence of feedbacks $(h_{t,i})_{t\in\mathbb{N}}$ for prediction i.*

To relate the non-delayed performance of BASE and the regret of QPM-D, we need a few definitions. For any t, let $S_i(t)$ denote the number of feedbacks for prediction i that are received by the end of time instant t. Then the number of missing feedbacks for i when making a prediction at time instant t is $G_{i,t} = T_i(t-1) - S_i(t-1)$. Let $G^*_{i,n} = \max_{1\le t\le n} G_{i,t}$. Furthermore, for each $i \in \mathcal{A}$, let $T'_i(t')$ be the number of times algorithm BASE has predicted i while being queried t' times. Let n' denote the number of steps the inner algorithm BASE makes in n steps of the real IPM problem. Next we relate n and n', as well as the number of times QPM-D and BASE (in its simulated environment) make a specific prediction.

Lemma 5. *Suppose QPM-D is run for $n \ge 1$ time instants, and has queried* BASE *n' times. Then $n' \le n$ and*

$$0 \le T_i(n) - T'_i(n') \le G^*_{i,n}. \tag{3}$$

Proof. Since BASE can take at most one step for each feedback that arrives, and QPM-D has to make at least one step for each arriving feedback, $n' \le n$.

Now, fix a prediction $i \in \mathcal{A}$. If BASE, and hence, QPM-D, has not predicted i by time instant n, (3) trivially holds. Otherwise, let $t_{n,i}$ denote the last time instant (up to time n) when QPM-D predicts i. Then $T_i(n) = T_i(t_{n,i}) = T_i(t_{n,i} - 1) + 1$. Suppose BASE has been queried $n'' \le n$ times by time instant $t_{n,i}$ (inclusive). At this time instant, the buffer $Q[i]$ must be empty and BASE must be predicting i, otherwise QPM-D would not predict i in the real environment. This means that all the $S_i(t_{n,i}-1)$ feedbacks that have arrived before this time instant have been fed to the base algorithm, which has also made an extra step, that is, $T'_i(n') \ge T'_i(n'') = S_i(t_{n,i} - 1) + 1$. Therefore,

$$\begin{aligned} T_i(n) - T'_i(n') &\le T_i(t_{n,i} - 1) + 1 - (S_i(t_{n,i} - 1) + 1) \\ &\le G_{i,t_{n,i}} \le G^*_{i,n}. \end{aligned}$$

□

We can now give an upper bound on the expected regret of Algorithm 2.

Theorem 6. *Suppose the non-delayed* BASE *algorithm is used in QPM-D in a delayed stochastic IPM environment. Then the expected regret of QPM-D is upper-bounded by*

$$\mathbb{E}[R_n] \le \mathbb{E}\left[R_n^{\text{BASE}}\right] + \sum_{i=1}^{K} \Delta_i \mathbb{E}\left[G^*_{i,n}\right], \tag{4}$$

where $\mathbb{E}\left[R_n^{\text{BASE}}\right]$ is the expected regret of BASE *when run in the same environment without delays.*

When the delay τ_t is bounded by τ_{max} for all t, we also have $G^*_{i,n} \le \tau_{max}$, and $\mathbb{E}[R_n] \le \mathbb{E}\left[R_n^{\text{BASE}}\right] + O(\tau_{max})$. When the sequence of delays for each prediction is i.i.d. with a finite expected value but unbounded support, we can use Lemma 2 to bound $G^*_{i,n}$, and obtain a bound $\mathbb{E}\left[R_n^{\text{BASE}}\right] + O(\mathbb{E}[\tau_1] + \sqrt{\mathbb{E}[\tau_1]\log n} + \log n)$.

Proof. Assume that QPM-D is run longer so that BASE is queried for n times (i.e., it is queried $n - n'$ more times). Then, since $n' \le n$, the number of times i is predicted by the base algorithm, namely $T'_i(n)$, can only increase, that is, $T'_i(n') \le T'_i(n)$. Combining this with the expectation of (3) gives

$$\mathbb{E}[T_i(n)] \le \mathbb{E}[T'_i(n)] + \mathbb{E}\left[G^*_{i,n}\right],$$

which in turn gives,

$$\sum_{i=1}^{K} \Delta_i \mathbb{E}[T_i(n)] \le \sum_{i=1}^{K} \Delta_i \mathbb{E}[T'_i(n)] + \sum_{i=1}^{K} \Delta_i \mathbb{E}\left[G^*_{i,n}\right]. \tag{5}$$

As shown in Lemma 4, the reordered rewards and feedbacks $h'_{i,1}, h'_{i,2}, \ldots, h'_{i,T'_i(n')}, \ldots h'_{i,T_i(n)}$ are i.i.d. with the same distribution as the original feedback sequence $(h_{t,i})_{t\in\mathbb{N}}$. The base algorithm BASE has worked on the first $T'_i(n)$ of these feedbacks for each i (in its extended run), and has therefore operated for n steps in a simulated environment with the same reward and feedback distributions, but without delay. Hence, the first summation in the right hand side of (5) is in fact $\mathbb{E}\left[R_n^{\text{Base}}\right]$, the expected regret of the base algorithm in a non-delayed environment. This concludes the proof. □

4. UCB for the Multi-Armed Bandit Problem with Delayed Feedback

While the algorithms in the previous section provide an easy way to convert algorithms devised for the non-delayed case to ones that can handle delays in the feedback, improvements can be achieved if one makes modifications inside the existing non-delayed algorithms

while retaining their theoretical guarantees. This can be viewed as a "white-box" approach to extending online learning algorithms to the delayed setting, and enables us to escape the high memory requirements of black-box algorithms that arises for both of our methods in the previous section when the delays are large. We consider the stochastic multi-armed bandit problem, and extend the UCB family of algorithms (Auer et al., 2002; Garivier & Cappé, 2011) to the delayed setting. The modification proposed is quite natural, and the common characteristics of UCB-type algorithms enable a unified way of extending their performance guarantees to the delayed setting (up to an additive penalty due to delays).

Recall that in the stochastic MAB setting, which is a special case of the stochastic IPM problem of Section 3.2, the feedback at time instant t is $h_t = r(a_t, b_t)$, and there is a distribution ν_i from which the rewards of each prediction i are drawn in an i.i.d. manner. Here we assume that the rewards of different predictions are independent of each other. We use the same notation as in Section 3.2.

Several algorithms devised for the non-delayed stochastic MAB problem are based on upper confidence bounds (UCBs), which are optimistic estimates of the expected reward of different predictions. Different UCB-type algorithms use different upper confidence bounds, and choose, at each time instant, a prediction with the largest UCB. Let $B_{i,s,t}$ denote the UCB for prediction i at time instant t, where s is the number of reward samples used in computing the estimate. In a non-delayed setting, the prediction of a UCB-type algorithm at time instant t is given by $a_t = \text{argmax}_{i \in \mathcal{A}} \, B_{i,T_i(t-1),t}$. In the presence of delays, one can simply use the same upper confidence bounds only with the rewards that are observed, and predict

$$a_t = \text{argmax}_{i \in \mathcal{A}} \, B_{i,S_i(t-1),t} \tag{6}$$

at time instant t (recall that $S_i(t-1)$ is the number of rewards that can be observed for prediction i before time instant t). Note that if the delays are zero, this algorithm reduces to the corresponding non-delayed version of the algorithm.

The algorithms defined by (6) can easily be shown to enjoy the same regret guarantees compared to their non-delayed versions, up to an additive penalty depending on the delays. This is because the analyses of the regrets of UCB algorithms follow the same pattern of upper bounding the number of trials of a suboptimal prediction using concentration inequalities suitable for the specific form of UCBs they use.

As an example, the UCB1 algorithm (Auer et al., 2002) uses UCBs of the form $B_{i,s,t} = \hat{\mu}_{i,s} + \sqrt{2\log(t)/s}$, where $\hat{\mu}_{i,s} = \frac{1}{s}\sum_{t=1}^{s} h'_{i,t}$ is the average of the first s observed rewards. Using this UCB in our decision rule (6), we can bound the regret of the resulting algorithm (called Delayed-UCB1) in the delayed setting:

Theorem 7. *For any $n \geq 1$, the expected regret of the Delayed-UCB1 algorithm is bounded by*

$$\mathbb{E}\left[R_n\right] \leq \sum_{i:\Delta_i>0} \left[\frac{8\log n}{\Delta_i} + 3.5\Delta_i\right] + \sum_{i=1}^{K} \Delta_i \mathbb{E}\left[G^*_{i,n}\right].$$

Note that the last term in the bound is the additive penalty, and, under different assumptions, it can be bounded in the same way as after Theorem 6. The proof of this theorem, as well as a similar regret bound for the delayed version of the KL-UCB algorithm (Garivier & Cappé, 2011) can be found in the extended version of the paper (Joulani et al., 2013).

5. Conclusion and future work

We analyzed the effect of feedback delays in online learning problems. We examined the partial monitoring case (which also covers the full information and the bandit settings), and provided general algorithms that transform forecasters devised for the non-delayed case into ones that handle delayed feedback. It turns out that the price of delay is a multiplicative increase in the regret in adversarial problems, and only an additive increase in stochastic problems. While we believe that these findings are qualitatively correct, we do not have lower bounds to prove this (matching lower bounds are available for the full information case only).

It also turns out that the most important quantity that determines the performance of our algorithms is G^*_n, the maximum number of missing rewards. It is interesting to note that G^*_n is the maximum number of servers used in a multi-server queuing system with infinitely many servers and deterministic arrival times. It is also the maximum deviation of a certain type of Markov chain. While we have not found any immediately applicable results in these fields, we think that applying techniques from these areas could lead to an improved understanding of G^*_n, and hence an improved analysis of online learning under delayed feedback.

6. Acknowledgements

This work was supported by the Alberta Innovates Technology Futures and NSERC.

References

Agarwal, Alekh and Duchi, John. Distributed delayed stochastic optimization. In Shawe-Taylor, J., Zemel, R.S., Bartlett, P., Pereira, F., and Weinberger, K.Q. (eds.), *Advances in Neural Information Processing Systems 24 (NIPS)*, pp. 873–881, 2011.

Auer, Peter, Cesa-Bianchi, Nicolò, and Fischer, Paul. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2-3):235–256, May 2002.

Cesa-Bianchi, Nicolò and Lugosi, Gábor. *Prediction, Learning, and Games*. Cambridge University Press, New York, NY, USA, 2006. ISBN 0521841089.

Desautels, Thomas, Krause, Andreas, and Burdick, Joel. Parallelizing exploration-exploitation tradeoffs with gaussian process bandit optimization. In *Proceedings of the 29th International Conference on Machine Learning (ICML)*, Edinburgh, Scotland, UK, 2012. Omnipress.

Dudik, Miroslav, Hsu, Daniel, Kale, Satyen, Karampatziakis, Nikos, Langford, John, Reyzin, Lev, and Zhang, Tong. Efficient optimal learning for contextual bandits. In *Proceedings of the 27th Conference on Uncertainty in Artificial Intelligence (UAI)*, pp. 169–178, Corvallis, Oregon, 2011. AUAI Press.

Garivier, Aurélien and Cappé, Olivier. The KL-UCB algorithm for bounded stochastic bandits and beyond. In *Proceedings of the 24th Annual Conference on Learning Theory (COLT)*, volume 19, pp. 359–376, Budapest, Hungary, July 2011.

Joulani, Pooria, György, András, and Szepesvári, Csaba. Online learning under delayed feedback. Extended version of a paper submitted to ICML-2013, 2013. URL http://webdocs.cs.ualberta.ca/~pooria/publications/DelayedFeedback-ICML2013-Extended.pdf.

Langford, John, Smola, Alexander, and Zinkevich, Martin. Slow learners are fast. In Bengio, Y., Schuurmans, D., Lafferty, J., Williams, C. K. I., and Culotta, A. (eds.), *Advances in Neural Information Processing Systems 22*, pp. 2331–2339. 2009.

Li, Lihong, Chu, Wei, Langford, John, and Schapire, Robert E. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th International Conference on World Wide Web (WWW)*, pp. 661–670, New York, NY, USA, 2010. ACM.

Mesterharm, Chris J. On-line learning with delayed label feedback. In Jain, Sanjay, Simon, HansUlrich, and Tomita, Etsuji (eds.), *Algorithmic Learning Theory*, volume 3734 of *Lecture Notes in Computer Science*, pp. 399–413. Springer Berlin Heidelberg, 2005.

Mesterharm, Chris J. *Improving on-line learning*. PhD thesis, Department of Computer Science, Rutgers University, New Brunswick, NJ, 2007.

Neu, Gergely, György, András, Szepesvári, Csaba, and Antos, András. Online markov decision processes under bandit feedback. In Lafferty, J., Williams, C. K. I., Shawe-Taylor, J., Zemel, R.S., and Culotta, A. (eds.), *Advances in Neural Information Processing Systems 23 (NIPS)*, pp. 1804–1812, 2010.

Weinberger, Marcelo J. and Ordentlich, Erik. On delayed prediction of individual sequences. *IEEE Transactions on Information Theory*, 48(7):1959–1976, September 2002.