Appendix for The Queue Method: Handling Delay, Heuristics, Prior Data, and Evaluation in Bandits

Travis Mandel[1], **Yun-En Liu**[1], **Emma Brunskill**[2], **and Zoran Popović**[1]
[1]Center for Game Science, Computer Science & Engineering, University of Washington
[2]School of Computer Science, Carnegie Mellon University
{tmandel, yunliu, zoran}@cs.washington.edu, ebrun@cs.cmu.edu

Stochastic Delayed Bandits

Theorem 1. *For algorithm 1 in the paper with any choice of procedure* GETSAMPLINGDIST *and any online bandit algorithm* BASE*,*

$$\mathbb{E}\left[R_T\right] \leq \mathbb{E}\left[R_T^{\text{BASE}}\right] + \sum_{i=1}^{N} \Delta_i \, \mathbb{E}\left[S_{i,T}\right] \tag{1}$$

where $S_{i,T}$ is the number of elements pulled for arm i by time T, but not yet shown to BASE*.*

Proof. Let T' be the number of times we have updated BASE. Note that since the samples fed to BASE were drawn iid according to the arm distributions and given on the arms it requested, the regret on these samples is $R_{T'}$. Clearly $T' \leq T$ since for every sample we give BASE, we must have correspondingly taken a step in the true environment[1]. So the expected regret of only those steps in which we update BASE can be upper bounded by

$$\mathbb{E}\left[R_T^{\text{BASE}}\right]. \tag{2}$$

Now we must consider those timesteps for which we requested a sample but have not given it to BASE by time T. The number of such samples from arm i is exactly $S_{i,T}$. So the regret on these timesteps is equal to

$$\sum_{i=1}^{N} \sum_{j=1}^{S_{i,T}} \rho_{i,j} \tag{3}$$

Where $\rho_{i,j}$ denotes the regret of the j^{th} sample not passed to BASE for arm i. Taking the expectation gives us:

$$\sum_{i=1}^{N} \mathbb{E}\left[\sum_{j=1}^{S_{i,T}} \rho_{i,j}\right] \tag{4}$$

By Wald's equation this is:

$$= \sum_{i=1}^{N} \mathbb{E}\left[S_{i,T}\right] \mathbb{E}\left[\rho_{i,j}\right] \tag{5}$$

$$= \sum_{i=1}^{N} \mathbb{E}\left[S_{i,T}\right] \Delta_i \tag{6}$$

Combining this with equation (2) gives us the required bound. □

Theorem 2. *For SDB,*

$$\mathbb{E}\left[R_T\right] \leq \mathbb{E}\left[R_T^{\text{BASE}}\right] + N\tau_{max} \tag{7}$$

in the online updating setting, and

$$\mathbb{E}\left[R_T\right] \leq \mathbb{E}\left[R_T^{\text{BASE}}\right] + 2N\tau_{max} \tag{8}$$

in the batch updating setting.[2]

Proof. First note that on all timesteps $B_t \leq \tau_{max}$, since $B_1 = 1$ and $\tau_{max} \geq 1$ and for all $t > 1$, B_t is equal to the empirical max delay.

We will now show that in the online updating case, $S_{i,t}$ when running SDB never exceeds τ_{max}. Clearly $S_{i,0} = 0$ and so the bound holds in the base case. Assume $S_{i,t} \leq \tau_{max}$ at t, and we will prove it for $t+1$. Let I_t denote the value of variable I in the SDB algorithm at time t. For arms $j \neq I_t$,the algorithm sets $p_{jt} = 0$ if $S_{jt} \geq B_t$, and since $B_t \leq \tau_{max}$ we cannot add samples such that $S_{j,t+1} > \tau_{max}$. So only $S_{I_t,t+1}$ can be greater than τ_{max}. However, we know Q_{I_t} must be empty, or

[1]We also make the trivial assumption that the provided regret bound monotonically increases with T.

[2]One small technical comment: Note that $\tau_{max} = 1$ refers to the case of no delay, since the arm pulled at time t is observed at time $t + 1$. It appears that Joulani et. al. 2013 used $\tau_{max} = 0$ to denote no delay, so there is an off-by-one difference in τ_{max} between our papers. This difference simply results in the SDB bound $\mathbb{E}\left[R_T^{\text{BASE}}\right] + \sum_i \Delta_i * c\tau_{max}$ becoming $\mathbb{E}\left[R_T^{\text{BASE}}\right] + \sum_i \Delta_i * c(\tau_{max} + 1)$ if we use the definition of Joulani et al., so in the worst case the online updating regret bound of SDB differs by at most N from that of QPM-D. This very small difference in the bounds could be eliminated by simply modifying the update of B in SDB to calculate maximum delay minus one instead of maximum delay, with a negligible effect on the performance of the algorithm. The only remaining issue would be that the $+N$ term would remain in the no-delay case since B is initialized to 1, however, the regret bounds would be the exactly same in the case of nonzero online updating delay.

variable I could not be set to point to it. But if we have more than τ_{max} samples assigned to I_t and not yet observed, one must have been assigned more than τ_{max} steps ago, meaning the delay must be greater than τ_{max}, a contradiction. So $S_{I_t,t+1}$ also cannot be greater than τ_{max}.

So we have shown that $S_{i,t} \leq \tau_{max}$ for all i and t in the online updating case. Plugging into Theorem 1 and observing that $\Delta_i \leq 1$ gives us the stated bound.

Now we will show $S_{i,t} \leq 2\tau_{max}$ in the batch update case. Note that in this case we refer to timesteps from the perspective of batches, where after every batch the timestep increments. Let B_t be the value of SDB variable B at time t.

Assume $S_{i,t} \leq 2\tau_{max}$ at t, and we will prove it for $t+1$.

For arm I_t, Q_{I_t} must be empty before the batch begins, so $S_{I_t,t} = 0$ since we have assumed all rewards from each batch are observed upon its completion. Since the batch size cannot be larger than τ_{max} it follows immediately that $S_{I_t,t+1} \leq \tau_{max} \leq 2\tau_{max}$. We now consider arms $i \neq I_t$.

In the case where $S_{i,t} \geq B_t$ and $i \neq I_t$, SDB sets $p_{i,t} = 0$, so $S_{i,t+1} = S_{i,t} \leq 2\tau_{max}$.

If $i \neq I_t$ and $S_{i,t} < B_t$, we know $B_t < \tau_{max}$, so $S_{i,t} < \tau_{max}$. So since the batch size cannot be larger than τ_{max}, $S_{i,t+1} \leq S_{i,t} + \tau_{max} < 2\tau_{max}$.

So we have shown that $S_{i,t} \leq 2\tau_{max}$ for all i and t in the batch updating case. Plugging into Theorem 1 and observing that $\Delta_i \leq 1$ gives us the stated bound. □

Lemma 1. *In the mixed case where we update in batches, but not all elements are guaranteed to return before the end of the batch,*

$$\mathbb{E}[R_T] \leq \mathbb{E}\left[R_T^{\text{BASE}}\right] + N(\tau_{max} + 2\beta_{max}) \quad (9)$$

for SDB, where β_{max} is the maximum size of a batch.

Proof. Observe that in this setting, while τ_{max} is the maximum delay of a sample before it returns, the maximum delay before we can process it is β_{max} + τ_{max}, since there are at most β_{max} steps between when a sample returns and when we can process it.

Let t refer to the timesteps on which updates occur. Let I_t denote the value of variable I in the SDB algorithm at time t, and similarly for B.

Clearly $S_{i,0} = 0$ and so the bound holds in the base case. Assume $S_{i,t} \leq 2\beta_{max} + \tau_{max}$ at time t, and we will prove it for $t+1$.

In the case $i = I_t$, we know that Q_{I_t} is empty at time t. Therefore, $S_{I_t,t} \leq \tau_{max}$ since if we have more than τ_{max} samples assigned to I_t and not yet observed, one must have been assigned more than τ_{max} steps ago, meaning the delay must be greater than τ_{max}, a contradiction. The most we can put in before the next update is β_{max}, so $S_{I_t,t+1} \leq \beta_{max} + \tau_{max} \leq 2\beta_{max} + \tau_{max}$.

In the case where $i \neq I_t$ and $S_{i,t} \geq B_t$, SDB sets $p_{i,t} = 0$, so $S_{i,t+1} = S_{i,t} \leq 2\beta_{max} + \tau_{max}$.

If $i \neq I_t$ and $S_{i,t} < B_t$, we know $B_t < \tau_{max} + \beta_{max}$, so $S_{i,t} < \tau_{max} + \beta_{max}$. So since the batch size cannot be larger than β_{max}, $S_{i,t+1} \leq S_{i,t} + \beta_{max} < \tau_{max} + 2\beta_{max}$.

So we have shown that $S_{i,t} \leq \tau_{max} + 2\beta_{max}$ for all i and t. Plugging into Theorem 1 and observing that $\Delta_i \leq 1$ gives us the stated bound. □

Comparison to prior work

Joulani et al. 2013 proposed a closely related algorithm QPM-D. If we extend their online updating analysis to the batch updating cases, their algorithm has a regret bound with a an additive term of $N\tau_{max}$ in the online updating and batch updating settings and $N(\tau_{max} + \beta_{max})$ in the mixed setting. Hence we see that SDB matches the bound of QPM-D in the online updating setting, but the additive term worsens by at most a factor of two when updates come in batches. Creating an algorithm that retains the bound of QPM-D in the batch case but also retains most of the empirical benefit of SDB is an important direction for future work.

Delay with Online Updating results

Due to space limitations, we were unable to include both the batch updating and online updating results in our paper.

The online updating case is relevant in many situations, and the comparison between SDB and QPM-D is a bit fairer because both possess the same theoretical guarantees in this case. Below we repeat the same experiments as in the batch case, the only difference being that each algorithm can update its distribution after each step instead of in batches.[3]

Figures 1a-1f show the results running SDB in the online updating case in a variety of simulations (see paper for explanations of the environments used). Figure 2 gives the results on actual data using our unbiased queue-based estimator. We see much the same results as we did in the batch updating setting, showing good empirical performance in a variety of scenarios. One notable difference is that in the UCB figures (1d and 1e) we see that SDB has essentially "smoothed" out performance, eliminating the the troughs of very poor performance. The reason for this is that in the online case, if the heuristic chooses an arm to pull, SDB will initially allow it to put full probability mass on that arm, but then smoothly decrease the amount of probability it is allowed place on that arm, shift the remainder to the

[3]The distribution of Uniform, QPM-D, UCB, UCB-Strict, and UCB-Discard does not change if new data is not observed, so their performance should be the same as in the batch updating setting. Therefore, we did not re-run those algorithms.

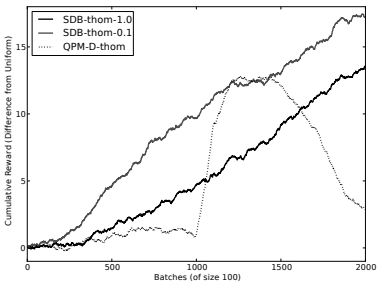


Figure 2: Results (using our queue-based estimator) on educational game data. SDB sees major improvements by leveraging the heuristic.

arm(s) recommended by BASE. Hence instead of just pulling one arm per batch, in an online updating setting SDB can spread mass between multiple arms, even given deterministic black-box algorithms.

Sample-Efficient Unbiased Offline Bandit Evaluation

Theorem 3. *Queue Replay Evaluation estimates are unbiased conditioned on the fact that the algorithm produces a sequence of actions for which we issue estimates. Specifically,*

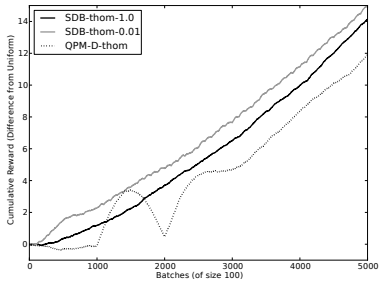
$$\mathbb{E}\left[r_t | s \in U_t\right] = \sum_{s'=\{i,\ldots,j\} \in U_t} p(s'|s' \in U_t, \theta)\mu_j$$

.

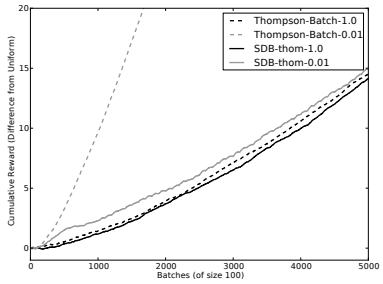
Proof. The proof builds on Lemma 4 in (Joulani, Gyorgy, and Szepesvari 2013). We can prove this by induction: At the initial step, the probability of the sequence and the expected value of the estimate at time 0 are trivially the same as in the real environment. Now, at time t, we know that the probability of seeing any past sequence of samples s'' is $p(s''|s'' \in U_{t-1}, \theta)$, and we also know all $t-1$ past estimates were issued with the correct prediction. So take any one of those sequences s'' (of length $t-1$). Now, the distribution over states of the algorithm given the past sequence of pulls is identical to the true system, since the state of the algorithm is defined by the past sequence of pulls and the rewards, and rewards were drawn iid from each arm. Therefore, $\hat{p}(j \text{ pulled}|s'') = p(j \text{ pulled}|s'')$, where $\hat{p}$ denotes the probability when running the queue-based evaluator. Therefore the probability of extending any $s'' \in U_{t-1}$ to any t-length sequence s' is the same as in the real environment, that is $\hat{p}(s'|s'' \in U_{t-1}) = p(s'|s'' \in U_{t-1})$. Therefore the marginal distribution $\hat{p}(s'|s'' \in U_{t-1}, s' \in U_t) = p(s'|s'' \in U_{t-1}, s' \in U_t)$. Now, given $s' \in U_t$, $s'' \in U_{t-1}$ since we must have not hit the end of the queue on step $t-1$ as well, so $\hat{p}(s'|s' \in U_t) = p(s'|s' \in U_t)$. Finally, if we pull arm j, the result is iid, so the expectation of r_t is μ_j. □

References

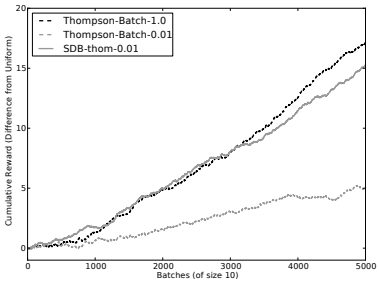
Joulani, P.; Gyorgy, A.; and Szepesvari, C. 2013. Online learning under delayed feedback. In *Proceedings of The 30th International Conference on Machine Learning*, 1453–1461.



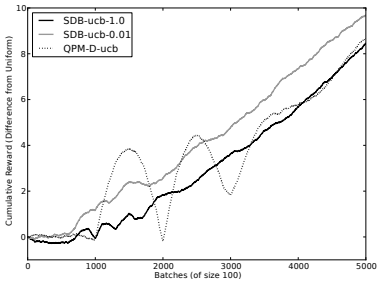
(a) Comparing to QPM-D with Thompson Sampling as the black-box algorithm.



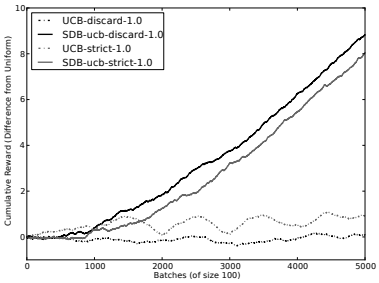
(b) Comparing to heuristics with Thompson Sampling as the black-box algorithm.



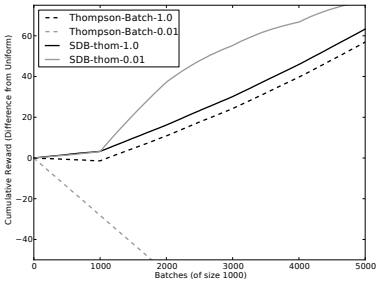
(c) An example where the heuristic Thompson-Batch-0.01 performs worse.



(d) Comparing to QPM-D using UCB as the black-box algorithm.



(e) Two example algorithms that perform poorly when handed all samples but well inside SDB.



(f) An example of a case where handing a prior dataset from a poor arm hurts Thompson-Batch but not SDB.

Figure 1: Simulation Results. SDB-thom-X refers to running SDB with Thompson-1.0 as the BASE algorithm and Thompson-X as the HEURISTIC algorithm, and likewise for UCB.