EE194/290S: ML for Sequential Decision under Uncertainty **Fall 2018**

Lecture 11: Adaptation 1

Lecturer: Anant Sahai/Vidya Muthukumar *Scribes: Seth Park, Kaylee Burns*

In this note, we recap The Doubling Trick and prove that we can achieve the same regret bound as we did when we knew the time horizon. This is the first in a series of adaptations we will discuss. At the end, we will introduce a second adaptation, where we incorporate information we've learned about the environment into our learning rate update procedure.

11.1 Recap and Reflection

Setting the Learning Rate

In previous notes, we assumed that we knew the time horizon, which allowed us to set the regularizer just right. Recall that the intuition for setting each of these learning rates was that we need to get to the optimum in $O\sqrt{T}$ time. This is more apparent in the learning rate for Online Gradient Descent, which we describe below.

For example, our learning rate for **Multiplicative Weights Updates** for discrete prediction with Hamming Loss was:

$$\eta^*_{mw} = \sqrt{\frac{\ln k}{l_{\max} T}}$$

Where k is the alphabet size, T is the time horizon, and $l_{\max}$ is the maximum size of the loss.

For **Online Gradient Descent**, this generalized to

$$\eta^*_{ogd} = \frac{B}{G\sqrt{2T}}$$

Where B is the size of the ball from which we're allowed to select weights: $W = \{u \mid \|u\| \leq B\}$. G constrains the "power" of our adversary; it is the maximum norm of the gradient. Because we know our gradients are never bigger than a certain amount, we can use this value to normalize our learning rate.

Can we relate how these learning rates were motivated? To get everywhere we need to go, we scale each learning rate by the how big we know our solution can be, the alphabet size in the former and the permissible region for weights in the latter. We also want to ensure that we normalize our updates by the maximum value of the loss, so we divide by $l_{\max}$ and G.

Coping with Unknown Time Horizons

The learning rates above rely on knowing how long our algorithm will be running. This assumption can be unreasonable in practice. In the last note, we described a strategy for

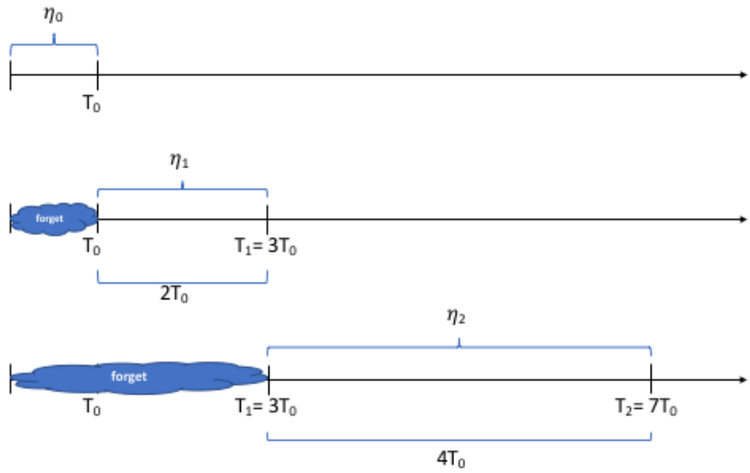


Figure 11.1: The Doubling Trick.

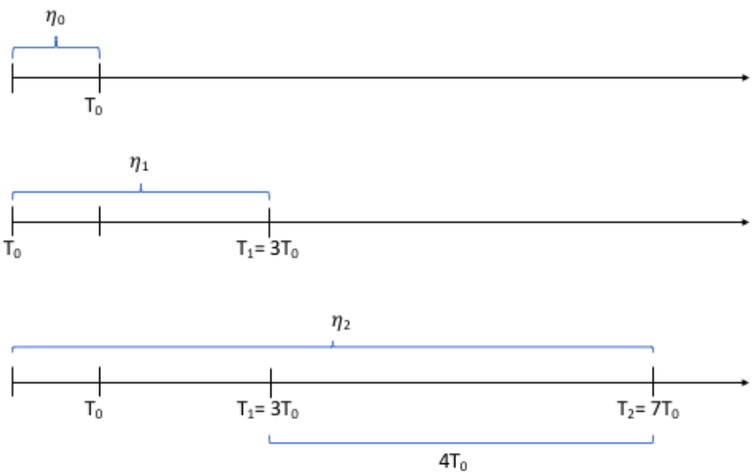


Figure 11.2: The Doubling Trick Without Forgetting.

coping with unknown time horizons by running our learning algorithm in independent intervals of a doubling T. *The Doubling Trick* is described in detail in Figure 11.1, Figure 11.2, and Algorithm 1. Run-Algorithm could be Multiplicative Weights or Follow the Perturbed Leader.

Algorithm 1 The Doubling Trick

1: Initialize T_0.
2: For $i \in \{0 \dots n\}$
3: $\quad T_i \leftarrow 2^i T_0$
4: $\quad$ Run-Algorithm(T_i)

The rest of this note will provide a bound on the regret accumulated by The Doubling Trick and conclude with an introduction to the next adaptation: sensitivity to the adversarial nature of the environment.

11.2 Bounding the Regret of The Doubling Trick

General Proof of Regret Bound

To guide our proof, we'll reflect on the generic strategy for proving regret bounds. In our proofs of regret so far, we bound the cumulative regret by the approximation error and estimation error then strive to balance the two terms by the end of the learning period.

$$\text{Cumulative Regret} \leq \text{Approximation Error} + \text{Estimation Error}$$

Recall the differences between error due to approximation and estimation:

Approx. Error	Est. Error
- Paid once	- Pay as you go
- In FTPL, size of the perturbation	- In FTPL, sum of all leader changes
- In ski example, buying skis	- In ski example, modified rental

How does this change when we don't know T? Now, we don't know how much estimation error we will pay over time, so we won't be able to pay the right approximation error up front. To cope with this, we balance as we go, paying approximation error once our estimation budget is exceeded. We pay approximately what we have paid so far or twice our last payment.

Can we achieve the same cumulative regret bound we get using the Doubling Trick? For the ease of analysis, we first prove for the case when we throw away the past and start fresh at each time segment.

Regret of The Doubling Trick

Because the length of our interval increases by a power of two, the amount of regret we accumulate also increases by a power of two. So, the sum of our costs is a geometric series, which is dominated by the final term asymptotically. We detail this below.

Proof[1]**:** The regret of The Doubling Trick is $O\sqrt{T}$.

Suppose $T_i = 2^i$ and $T_{true} = \sum_{i=0}^{n} T_i$. $CR_A(B)$ is the cumulative regret accumulated over time B with a learning rate set for a time horizon of A.

$$CR_{\text{total}} \leq \sum_{i=0}^{\text{n}} CR_{T_i}(T_i) \tag{11.1}$$

$$= \sum_{i=0}^{n} c * \sqrt{2^i} \tag{11.2}$$

$$= c \sum_{i=0}^{n} 2^{\frac{i}{2}} \tag{11.3}$$

$$\approx c * \frac{2^{\frac{n+1}{2}} - 1}{\sqrt{2} - 1} \tag{11.4}$$

$$\in O\sqrt{T_{true}} \tag{11.5}$$

In step 11.1 we use the fact that the total cumulative regret is less than the sum of the cumulative regret bound per interval. In step 11.2, we use the fact that the cumulative regret bound for an interval of length T is $\mathcal{O}(\sqrt{T})$ for Multiplicative Weights and Follow-The-Perturbed-Leader. We include a generic proportionality constant, c, which incurred from the approximation error. Observe that the expression in step 11.3 is a geometric sum with $a = 2$ and $r = \sqrt{2}$.

[1]The intention of both proofs in this section is to provide substance to our intuition. Therefore, they lack the formality of standard proofs.

Step 11.2 of the preceding proof assumes that we throw away information from all previous intervals. We would have to throw away all of our weights at the start of a new interval. This seems silly. *If we keep our weights at each time step, can we achieve the same asymptotic bound?*

A very coarse analysis proves that we can. We will use the fact that the regret in an interval T_i can't be more than the regret of running the algorithm from the start of T_0 to the end of T_i.

Proof: The regret of The Doubling Trick is $O\sqrt{T}$ even if we keep weights from previous intervals.

Suppose $T_i = 2^i$ and $T_{true} = \sum_{i=0}^{n} T_i$. $CR_A(B)$ is the cumulative regret accumulated over time B with a learning rate set for a time horizon of A.

$$CR_{\text{total}} \leq \sum_{i=0}^{\text{n}} CR_{T_i}\left(\sum_{j=0}^{i} T_j\right) \tag{11.6}$$

$$= \sum_{i=0}^{n} c * \sqrt{\sum_{j=0}^{i} 2^j} \tag{11.7}$$

$$\approx \sum_{i=0}^{n} c * 2^{\frac{i+1}{2}} \tag{11.8}$$

$$= \sum_{i=0}^{n} c\sqrt{2} * 2^{\frac{i}{2}} \tag{11.9}$$

$$\approx c\sqrt{2}\frac{\sqrt{2^{n+1}} - 1}{\sqrt{2} - 1} \tag{11.10}$$

$$\in O\sqrt{T_{true}} \tag{11.11}$$

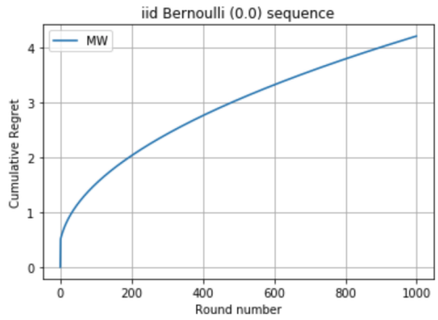
In step 11.6 we note that the cumulative regret is at most the regret per interval. The regret in each interval is less than or equal to the regret that would be accumulated by the algorithm from the beginning to the end of that interval. We apply the closed form of the geometric sum once in step 11.8 and again in step 11.10.

11.3 Be Greedier: How Mean was my World?

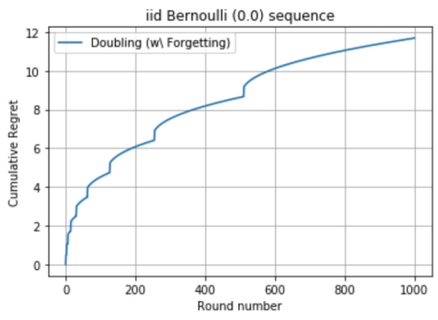
In the previous sections, we have shown that the cumulative regret bound in the Doulbing Trick is still sublinear even without knowing the exact T. The proofs relied on the assumptions about the upper bound of the estimation error term. In other words, we assume the worst-case of the w's . However, in the online learning setting where we execute the algorithm as we go, it is important to note that we are aware of the actual losses. This means that we do not need to rely on the bounds of the estimation error term. Let's take a look at the mixability gap:

$$\text{Estimation Term} \leq \sum_{t=1}^{T}(\sum_{j=1}^{k} w_t[j] \cdot l_t[j]) + \frac{1}{\eta}\ln(\sum_{j=1}^{k} w_t[j] \cdot e^{-\eta l_t[j]}) \tag{11.12}$$

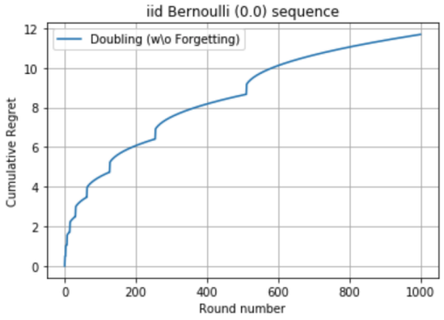
Recall that the first term of the right hand side of the inequality 11.12 indicates the true expected error that we would have incurred. The second term indicates a moment generating function which is an expectation of the random variables and can be bounded because the loss is bounded. However, these terms are not actually just random variables and their expected value; we actually know what these terms are as we run our algorithm. What this means is that if the actual estimation error is not as bad as we thought, we may not necessarily have to overreact. In other words, if nature has not presented us with the worst-case



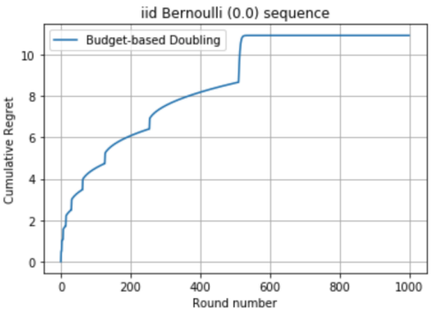
(a) Multiplicative Weight Knowing T



(b) Doubling Trick with Forgetting

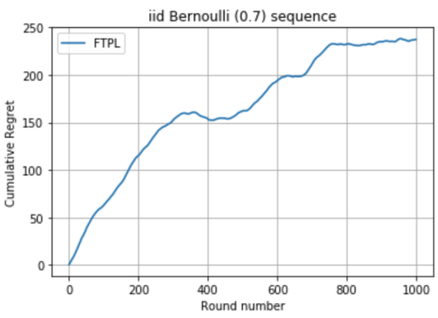


(c) Doubling Trick without Forgetting

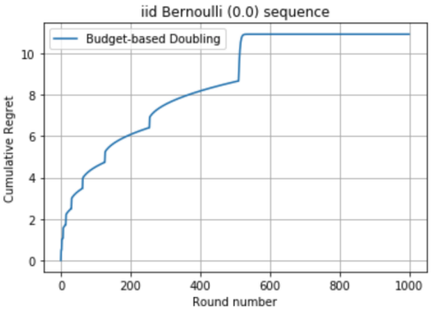


(d) Budget-based Doubling Trick

Figure 11.3: $X_i \sim$ Bernoulli(0)



(a) Follow-The-Perturbed-Leader



(b) Budget-based Doubling Trick

Figure 11.4: $X_i \sim$ Bernoulli(0.7)

scenario, then there is no reason to change the learning rate. *We should change our learning rate when the estimation error exceeds our approximation budget.*
Then what would happen if we do this? In the worst case, we would still have our sublinear bound on the cumulative regret. But sometimes we would do even better and achieve lower cumulative regret in which we saturate our bound as long as we can. Looking at some plots might illustrate this point better.

11.3.1 Example Plots

Suppose $X_i \sim$ Bernoulli(0) (i.e. the sequence is 0, 0, 0, ... 0). In Figure 11.3, we present plots of cumulative loss given timesteps for various algorithms. For Multiplicative Weights, Doubling Trick with Forgetting, and Doubling Trick without Forgetting, we see that the cumulative loss, although asymptotically bounded, increases over time. However, for the Budget-based Doubling Trick, once the weights being predicted are concentrated and confident, the mixability approaches to zero and the estimation error term will no longer exceed the approximation error term. This means that the learning rate is no longer decreased and the cumulative regret becomes bounded by a constant.

Now suppose $X_i \sim$ Bernoulli(0.7) where the sequence is stochastic, but has a clear winner. For FTPL in Figure 11.4, recall the intuition of why the regret plateaus. In the earlier phase of FTPL, the relative significance of the perturbation is high which in turn influences the

number of leader changes occurring. As the learning proceeds, however, a sequence indicating the clear winner will become more prominent and the influence of the perturbation on the number of leader changes will decrease. And as the number of leader changes converges, the regret bound will plateau as a result.
Similarly, during the Budget-based Doubling Trick, the number of leader changes will stay constant once the estimation error term converges to zero. As in the previous case, the cumulative regret approaches some constant term.