THEORETICAL MACHINE LEARNING COS 511

LECTURE #23 APRIL 26, 2016

LECTURER: ELAD HAZAN SCRIBE: MERT AL

In this lecture, we will cover the topic of reinforcement learning (RL), which can be described as the process of learning how to map states to actions in order to maximize some cumulative reward. To stay consistent with our formulations so far, we will define RL as the problem of minimizing cumulative loss, in other words, minimizing regret. We will put emphasis on regret minimization in Adversarial Deterministic Markov Decision Processes (ADMDP), where losses incurred for any set of actions taken by the player are deterministic.

We will first provide a formal definition of the general problem of reinforcement learning and show that it is not possible to achieve sublinear regret bounds for the general case. We will then introduce the necessary (and sufficient) conditions to guarantee asymptotically vanishing average regret (i.e. sublinear regret). Finally we will perform a reduction from ADMDP to online routing and relate it to the Multi-Armed Bandit (MAB) problem, so that we can describe an algorithm that achieves tight regret bounds.

1 Reinforcement Learning

In Reinforcement Learning, we have a set of states denoted by $\mathcal{V}$ and at each state v, we have possible actions denoted by $a(v)$. Each action is associated with a transition to another state $v' = \nu(v, a(v))$ as well as a loss (or reward), which we denote by $l(v, v')$. We can represent the state of the world in this setting with a directed graph $G = (\mathcal{V}, E)$, where states are represented by the vertices and possible actions from each state are represented by the edges.

The goal of learning in this setting is to minimize the long term regret. So the problem can be written as

$$\begin{aligned} \underset{a_1,a_2,...}{\text{minimize}} \quad & \sum_{t=0}^{\infty} l(v_t, v_{t+1})\beta^t \\ \text{where} \quad & v_{t+1} = \nu(v_t, a_t) \end{aligned}$$

and $\beta \leq 1$. A policy $\Pi = \{a_1, a_2, ...\}$ is a mapping from $\mathcal{V}$ to $\mathcal{V}$. The goal of reinforcement learning can also be described as finding the best policy Π^*, where

$$\Pi^* = \underset{a_1,a_2,...}{\text{argmin}} \sum_{t=0}^{\infty} l(v_t, v_{t+1})\beta^t$$

Finally we denote by $\Pi(v)$ the vertex reached from v by policy Π.

We will now consider a special case of Reinforcement Learning in which we do not know the costs associated with actions, and all we can observe is the loss we incur at time t.

2 Adversarial Deterministic Markov Decision Process

Consider a scenario, where we are trying to minimize regret under the following conditions:

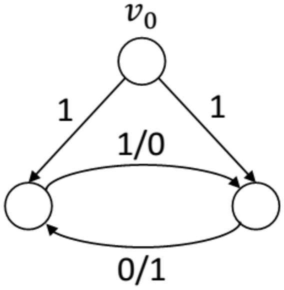
1. $l(v_t, v_{t+1})$ is only observable at time t.

2. $l_t : \mathcal{V} \to \mathbb{R}$ is adversarial, in other words losses associated with policies change within the set $[0\ 1]$ depending on the phase of the ADMDP.

Definition 2.1. Regret of the reinforcement learning algorithm is defined as:

$$R_T = E\left[\sum_{t=1}^{T} l_t(v_t, v_{t+1})\right] - \min_{\Pi} \sum_{t=1}^{T} l_t(\Pi^t(v_0), \Pi^{t+1}(v_0))$$

A key observation in this setting is the fact that every policy $\Pi : \mathcal{V} \to \mathcal{V}$ induces a cycle in the graph G. So what we really need to do in order to minimize regret is to find a cycle/phase combination that gives the minimum cumulative loss. Unfortunately there is no guarantee under the general setting that we will ever find such a combination, even after infinite time. Consider the following graph:



Example 2.2. In this example, adversarial loss has only two phases and shifts between 0 and 1 losses for the edges at the bottom. The player at the beginning has no idea which policy will give her minimum regret. If she takes the right path at the beginning, she enters a cycle where she continuously experiences 0 loss. Conversely if she takes the left path, she enters a cycle where she continuously experiences 1 loss. According to this graph the player may never enter the cycle with minimum loss, so asymptotically vanishing average regret is impossible to guarantee.

What prevents a player from finding the cycle with minimum regret in the above example is the fact that once an action is taken, a player has no chance to go back to the initial vertex and enter the same cycle with a different phase. Therefore, we conclude from this example that in order to guarantee $\frac{R_T}{T} \to 0$ as $T \to \infty$, we need strong connectivity in the graph. In other words, we need to be able to explore all phase/cycle combinations in the graph. For the regret guarantees we will establish, strong connectivity is also a sufficient condition.

Theorem 2.3. ***[Dekel, Ding, Koren,...]*** *The regret bound for ADMDP is $\Omega(T^{2/3})$, so it cannot be improved further [1].*

Theorem 2.4. *Let n be the number of states and m be the number of edges in a strongly connected Adversarial Deterministic Markov Decision Process. Then there exists an efficient algorithm with regret $O(n^2 m T^{2/3})$.*

Proof. We are able to explore all cycles with all phases in the graph. We will consider policies, which are essentially cycles whose lengths are less than or equal to $|\mathcal{V}|$.

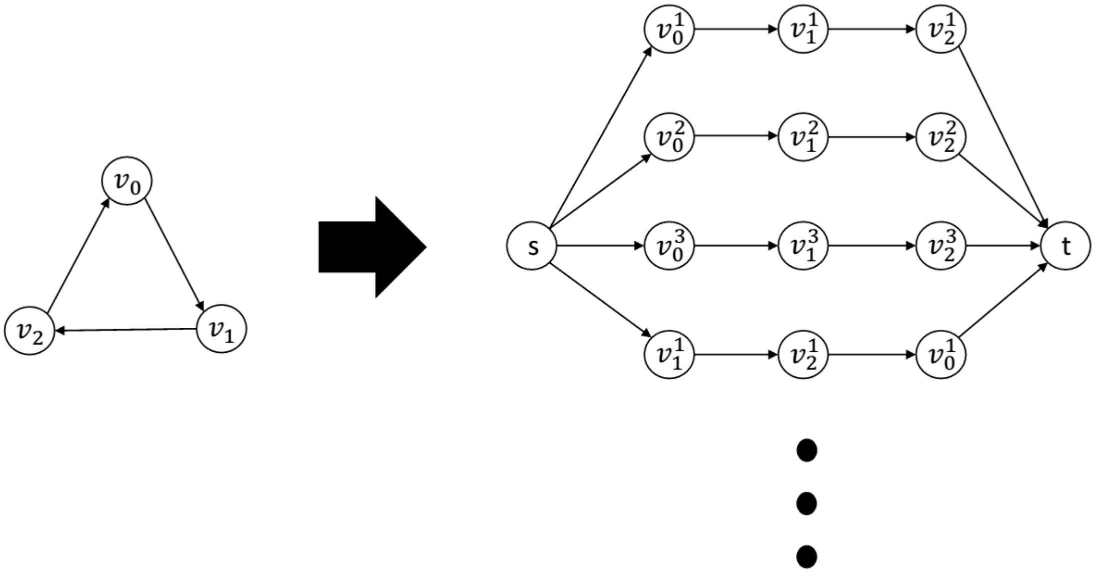
Definition 2.5. Period of a vertex v in graph G, denoted by $period(v)$ is the greatest common divisor of the lengths of all cycles starting at v.

Let G be a strongly connected graph, let u and v be two vertices of this graph. Then $\forall u, v \in G$, $period(u) = period(v) = period\ of\ the\ graph.$ Theorem 2.4 is true for all periods, but for simplicity we will assume that period of the graph is 1, and make use of the following theorem.

Theorem 2.6. *Let n be the number of states and m be the number of edges in a strongly connected graph G. If $period(G) = 1$, $\exists d \in \mathbb{N}$ such that $\forall s \geq d$, $\forall u, v \in G$, there exists a path from u to v of length exactly s. Furthermore $d \leq n(n-1)$.*

Theorem 2.6 allows us to transition between cycle/phase combinations with a number of actions that is upper bounded. Let p be the number of adversarial phases in the graph. Then with at most $d+p-1$ actions, we can reach any starting vertex with any phase from any vertex of the graph.

Now we shall perform a reduction from ADMDP to online routing. We may construct an online routing graph G' from the ADMDP graph G such that each path in G' corresponds to a starting vertex and phase in G. A path in this new graph should encompass all the cycles that start with a particular vertex and phase. We will demonstrate the construction of such a graph with a simple example.



Example 2.7. As you can see, even with a very simple graph G with 3 adversarial phases, G' ends up having many paths representing the possible policies. We would have to draw 9 paths for the graph on the right (G') to capture all the cycle and phase combinations in G starting at different vertices. Note that paths in G' are very simple because there aren't many possible cycles in G, e.g. there is only a single possible cycle that starts and ends with v_0^1.

Note that minimum regret for the online routing problem in G' corresponds to minimum regret for the reinforcement learning problem in G, except for the transition losses in RL. Thus we can relate RL to the MAB problem, where there is a cost associated with switching arms. Since transition cost in G is upper bounded by $n(n-1)+p-1$ due to Theorem 2.6, we can scale it down to 1 without loss of generality.

Lemma 2.8. *There exists an algorithm for the MAB problem with switching costs with regret bound $O(T^{2/3})$.*

Proof. We may divide time into blocks of length τ and run $EXP3$ algorithm on the blocks. Then if we denote the number of arms as N, regret of the algorithm is bounded as follows:

$$\begin{aligned}
Regret_T &\leq \tau Regret_{\frac{T}{\tau}}(EXP3) + \frac{T}{\tau} && (\frac{T}{\tau}\ blocks\ and\ transitions) \\
&\leq \tau\sqrt{\frac{T}{\tau} NlogN} + \frac{T}{\tau} && (Lemma\ 6.2) \\
&\leq T^{2/3}\sqrt{NlogN} + T^{2/3} && (Choice\ of\ \tau = T^{1/3}) \\
&= O(T^{2/3})
\end{aligned}$$

□

We can think of the paths in G' as arms. Then trying an arm in the MAB problem corresponds to trying cycles that start from a particular vertex and phase in ADMDP. As losses accumulated in MAB with switching costs and ADMDP directly relate to each other, we have successfully shown an algorithm for ADMDP with expected regret bound $O(T^{2/3})$, which according to Theorem 2.3 is the best we can do. $\square$

Factor n^2m in the regret bound given in Theorem 2.4 is a consequence of G' having $O(n^2m)$ edges. The curious reader may refer to [2] for a more detailed proof of this theorem.

References

[1] Ofer Dekel, Jian Ding, Tomer Koren, and Yuval Peres. Bandits with switching costs: T 2/3 regret. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*, pages 459–467. ACM, 2014.

[2] Ofer Dekel and Elad Hazan. Better rates for any adversarial deterministic mdp. In *Proceedings of The 30th International Conference on Machine Learning*, pages 675–683, 2013.