Learning Multiple Markov Chains via Adaptive Allocation

Mohammad Sadegh Talebi and Odalric-Ambrym Maillard

Inria Lille – Nord Europe

NeurIPS 2019

Problem

We are interested in learning transition matrices of multiple unknown Markov chains, under some error metric:

- with a finite budget n,
- using a single trajectory on each chain,
- and we wish to be competitive with an oracle, which is of some properties of the chains.

Motivation:

- Active exploration in MDPs, where one seeks to learn transition kernel of an unknown MDP from a single trajectory
- Active learning in rested Markov bandits

Model

K ergodic Markov chains are given, all defined on a finite state space $\mathcal{S}$ with cardinality S. For each chain k,

- P_k: transition matrix of k,
- π_k: stationary distribution of k, with $\min_x \pi_k(x) > 0$.
- $\gamma_{\mathsf{ps},k}$: (pseudo-)spectral gap of k.
- Introduce: $G_k(x) = \sum_{y\in\mathcal{S}} P_k(x,y)(1 - P_k(x,y))$.

Initially all chains are assumed to be non-stationary with arbitrary initial distributions.

- At each step $t \geq 1$, the learner samples a chain k_t, based on the past decisions and the observed states, and observes the state $X_{k_t,t}$.
- **Rested Setting:** The state of k_t evolves according to P_{k_t}. And $X_{k,t} = X_{k,t-1}$ for all $k \neq k_t$.

Some Definitions

- $T_{k,t}$: # of times chain k is selected up to time t
- $T_{k,x,t}$: # of times chain k is selected up to t, and it was in state x

Introduce empirical stationary distributions $\hat{\pi}_{k,t}$:

$$\hat{\pi}_{k,t}(x) := \frac{T_{k,x,n}}{T_{k,n}}, \quad \forall x \in \mathcal{S}$$

and $\widehat{P}_{k,t}$, α-smoothed estimators for P_k:

$$\widehat{P}_{k,t}(x,y) := \frac{\alpha + \sum_{t'=2}^{t} \mathbb{I}\{X_{k,t'-1} = x, X_{k,t'} = y\}}{\alpha S + T_{k,x,t}}, \quad \forall x, y \in \mathcal{S}$$

Note: *Laplace-smoothed* estimator when $\alpha = 1/S$.

Performance Measure

The loss of an algorithm $\mathcal{A}$, given budget n:

$$L_n(\mathcal{A}) := \max_{k\in[K]} \sum_{x\in\mathcal{S}} \hat{\pi}_{k,n}(x)\|P_k(x,\cdot) - \widehat{P}_{k,n}(x,\cdot)\|_2^2$$

The learner wishes to design a sequential allocation strategy to adaptively sample various MCs so that all transition matrices are learnt uniformly well w.r.t. loss L_n.

Let $\delta \in (0,1)$. For a given algorithm $\mathcal{A}$, under a loss function L, we wish to find $\varepsilon := \varepsilon(n,\delta)$ such that

$$\mathbb{P}\left(L_n(\mathcal{A}) \geq \varepsilon\right) \leq \delta\,.$$

Comparison with Other Losses

Alternative loss: $L'_n(\mathcal{A}) = \max_k \sum_x \|P_k(x,\cdot) - \widehat{P}_{k,n}(x,\cdot)\|_2^2$

- L' incurs a high loss for a part of the state space that is rarely visited, even though we have absolutely no control on the chain.
- When some state x is reachable with a very small probability, $T_{k,x,n}$ may be very small thus yielding a large L'_n for all algorithms, while it makes little sense to penalize an algorithm for such a "virtual" state.

Alternative loss: $L''_n(\mathcal{A}) = \max_k \sum_x \pi_k(x)\|P_k(x,\cdot) - \widehat{P}_{k,n}(x,\cdot)\|_2^2$

- When n is "small", $\hat{\pi}_{k,n}$ could differ significantly from π_k. The use of $\pi_k(x)$ does not seem reasonable as in a given sample path, the algorithm might not have visited x enough even though $\pi_k(x)$ is not small. Yet using $\widehat{\pi}_{k,n}(x)$ makes more sense as it accounts for the number of rounds the algorithm has actually visited x.

Static Allocation

Lemma

We have for any chain k: $T_{k,n} L_{k,n} \to_{T_{k,n}\to\infty} \sum_x G_k(x)$.

Now, consider an oracle policy $\mathcal{A}_{\text{oracle}}$, who is aware of $\sum_{x\in\mathcal{S}} G_k(x)$ for various chains. Using the above lemma and $\sum_{k\in[K]} T_{k,n} = n$, it would be asymptotically optimal to allocate $T_{k,n} = \eta_k n$ samples to chain k, where

$$\eta_k := \frac{1}{\Lambda}\sum_{x\in\mathcal{S}} G_k(x)\,, \quad \text{with} \quad \Lambda := \sum_{k\in[K]}\sum_{x\in\mathcal{S}} G_k(x)\,.$$

The corresponding loss would satisfy: $nL_n(\mathcal{A}_{\text{oracle}}) \to_{n\to\infty} \Lambda$.

Definition (Uniformly Good Algorithm)

An algorithm $\mathcal{A}$ is said to be uniformly good if, for any problem instance, it achieves the asymptotically optimal loss when n grows large; that is, $\lim_{n\to\infty} nL_n(\mathcal{A}) = \Lambda$ for all problem instances.

The **BA-MC** Algorithm

We present **BA-MC** (Bandit Allocation for Markov Chains):

- Designed based on the optimistic principle
- Easy to implement

BA-MC maintains a index function $b_{k,t}$ for each chain k at time t.

- Index $b_{k,t}$ is constructed as the UCB on the loss function $L_{k,t}$.
- After an initialization phase, it simply selects the chain with the largest index $b_{k,t}$ in each round t.

The BA-MC Algorithm

We present **BA-MC** (Bandit Allocation for Markov Chains), designed based on the optimism in face of uncertainty principle as in stochastic bandits.

BA-MC maintains an index function $b_{k,\cdot}$ for each chain k:

Index function for chain k

$$b_{k,t+1} = \frac{2\beta}{T_{k,t}} \sum_{x:T_{k,x,t}>0} \widehat{G}_{k,t}(x) + \frac{6.6\beta^{3/2}}{T_{k,t}} \sum_{x\in\mathcal{S}} \frac{T_{k,x,t}^{3/2}}{(T_{k,x,t}+\alpha S)^2} \sum_{y\in\mathcal{S}} \sqrt{\widehat{P}_{k,t}(I-\widehat{P}_{k,t})(x,y)}$$
$$+ \frac{28\beta^2 S}{T_{k,t}} \sum_{x:T_{k,x,t}>0} \frac{1}{T_{k,x,t}+\alpha S}$$

where $\beta := c\log\left(\left\lceil \frac{\log(n)}{\log(c)} \right\rceil \frac{6KS^2}{\delta}\right)$, with $c>1$ being an arbitrary choice (we choose $c=1.1$), and where $\widehat{G}_{k,t}(x) := \sum_x \widehat{P}_{k,t}(I-\widehat{P}_{k,t})(x,y)$.

The BA-MC Algorithm

The design of index $b_{k,\cdot}$ comes from the application of empirical Bernstein concentration for α-smoothed estimators (Lemma 4) to $L_{k,t}$.

$\Rightarrow b_{k,t+1} \geq L_{k,t}$ with high probability.

Algorithm 1 **BA-MC**– Bandit Allocation for Markov Chains

Input: Confidence parameter δ, budget n, state space $\mathcal{S}$;
Initialize: Sample each chain twice;
for $t = 2K+1, \ldots, n$ **do**
  Sample chain $k_t \in \mathrm{argmax}_k \, b_{k,t+1}$;
  Observe $X_{k,t}$, and update $T_{k,x,t}$ and $T_{k,t}$;
end for

Performance

Theorem (BA-MC, Generic Performance)

Let $\delta \in (0,1)$. Then, for any budget $n \geq 4K$, with probability at least $1-\delta$,

$$L_n \leq \frac{304 K S^2 \beta^2}{n} + \widetilde{\mathcal{O}}\Big(\frac{K^2 S^2}{n^2}\Big) .$$

The above bound holds even if the Markov chains $P_k, k \in [K]$ are reducible or periodic.

Performance

Introduce for any chain k:

$$H_k = \sum_{x \in \mathcal{S}} \pi_k(x)^{-1} \text{ and } \quad \underline{\pi}_k := \min_{x \in \mathcal{S}} \pi_k(x) > 0,$$

and recall $\Lambda = \sum_k \sum_x G_k(x)$ and $\eta_k = \frac{\sum_x G_k(x)}{\Lambda}$.

Theorem (BA-MC)

Let $\delta \in (0,1)$, and assume that $n \geq n_{\textit{cutoff}}$, where $n_{\textit{cutoff}} := K \max_k \left(\frac{300}{\gamma_{\textit{ps},k} \underline{\pi}_k} \log\left(\frac{2K}{\delta} \sqrt{\underline{\pi}_k^{-1}} \right) \right)^2$. Then, with probability at least $1 - 2\delta$,

$$L_n \leq \frac{2\beta\Lambda}{n} + \frac{C_0 \beta^{3/2}}{n^{3/2}} + \widetilde{\mathcal{O}}(n^{-2}),$$

where $C_0 := 150K\sqrt{S\Lambda \max_k H_k} + 3\sqrt{S\Lambda} \max_k \frac{H_k}{\eta_k}$.

Performance

Theorem (BA-MC, Asymptotic Performance)

We have $\limsup_{n\to\infty} nL_n = \Lambda$.

Three regimes depending on the budget n:

- Small-budget $(n \geq 4K)$: $L_n = \widetilde{\mathcal{O}}(\frac{KS^2}{n})$
- Larger-than-cutoff-budget $(n \geq n_{\text{cutoff}})$: $L_n = \widetilde{\mathcal{O}}(\frac{\Lambda}{n} + \frac{C_0}{n^{3/2}})$
- Asymptotic $(n \to \infty)$: $nL_n \to \Lambda$

$L_n(\mathcal{A}) - \frac{2\beta\Lambda}{n}$ may be thought of as the *pseudo-excess loss* of $\mathcal{A}$: when $n \geq n_{\text{cutoff}}$, the pseudo-excess loss under **BA-MC** vanishes at a rate $\widetilde{\mathcal{O}}(C_0 n^{-3/2})$, where C_0 is a problem-dependent quantity.

Connection to Active Learning

Closest to our problem is "active learning in bandits" [Antos et al. (2010), Carpentier et al. (2011), Carpentier et al. (2015)]:

- Loss $L_n = \max_k \mathbb{E}[(\mu_k - \hat{\mu}_{k,n})^2]$
- Regret $R_n = L_n - L_n^\star$

Algorithms with $R_n = \widetilde{\mathcal{O}}(n^{-3/2})$ are proposed in [Antos et al. (2010), Carpentier et al. (2011)].

Conclusion and Future Directions

Learning transition matrices of ergodic Markov chains was addressed:

- The notions of loss (for various distance functions)
- Characterization of a uniformly good algorithm
- Introduced **BA-MC**, whose (problem-dependent) excess-loss grows as $\widetilde{\mathcal{O}}(n^{-3/2})$

Future directions:.

- Lower bounds
- Extension to non-ergodic chains