Journal of Machine Learning Research () Submitted 4/00; Published 10/00

Contextual Markov Decision Processes

Assaf Hallak IFOGPH@GMAIL.COM
The Technion, Haifa, Israel

Dotan Di Castro DOTAN.DICASTRO@GMAIL.COM
Yahoo Labs, Haifa, Israel

Shie Mannor SHIE@EE.TECHNION.AC.IL
The Technion, Haifa, Israel

Editor: EWRL

Abstract

We consider a planning problem where the dynamics and rewards of the environment depend on a hidden static parameter referred to as the *context*. The objective is to learn a strategy that maximizes the accumulated reward across all contexts. The new model, called *Contextual Markov Decision Process* (CMDP), can model a customer's behavior when interacting with a website. The customer's behavior depends on gender, age, location, device, etc. Based on that behavior, the website objective is to *determine* customer characteristics, and to *optimize* the interaction between them. Our work focuses on one basic scenario–finite horizon with a small number of possible contexts. We suggest a family of algorithms with provable guarantees that learn the underlying models and the latent contexts, and optimize the CMDPs. Bounds are obtained for specific naive implementations, and extensions of the framework are discussed, laying the ground for future research.

1. Introduction

Markov Decision Processes (MDPs) are commonly used to describe dynamic behavior in multiple fields such as signal processing, robotics, advertising and queues management (Puterman, 2005). In many applications there are additional exogenous variables that affect the model, we refer to as the *context*. For example, the temporal behavior of sugar levels for diabetes patients is largely influenced by their age and gender. Since these context variables do not change within each measurement, incorporating them into the state creating a much larger MDP or POMDP seems faulty as it reduces the generalizing power of the model. Specifically, incorporating static features into the state forms distinct unconnected dynamic chains. As transition probability between states with different contexts is always zero, a more compact model would be separate transition matrices for each context instead of one double sized matrix.

A real world example for latent context learning is the problem of *identifying the user*. Consider a large content website. Such a website has two main activities: (a) suggesting relevant content to its users and (b) presenting alluring ads for profit. Current methodologies that determine the relevance of the content and the ads require the user profile: age, gender, income level, device, location, etc. Usually, in order to determine whether a certain user is revisiting the website, mechanisms such as (HTTP) cookies are used. But in many cases these mechanisms are insufficient. What if the website does not have any prior information about the user (also known as the *cold start problem*)? Can we learn the user's age or gender by observing his interaction with the website? More importantly, can

we take advantage of such clustering and tailor the policy to the user? In such cases, we model the interaction of a user as a *Markov Decision Process* where different users groups may be modeled and optimized according to their context. In on-line advertising, solutions to such optimization problem are highly valuable, where the correct identification of users leads to higher *click through rates* (CTRs). Hence, the ultimate goal is *on-line learning an optimal control when both the context, and the model's parameters are unknown*.

Our work's main contribution is presenting a general algorithm with provable guarantees for the finite horizon episodic contextual MDP setup. Considering a specific implementation, we provide regret analysis and empirical parametric sensitivity analysis. The reader should bear in mind the solutions suggested are preliminary and our focus is on presenting the problems along with their derived trade-offs, as well as setting the bar for future research.

2. Contextual Markov Decision Processes

We begin with defining a standard Markov Decision Process (MDP; Puterman 2005).

Definition 1 *(MDP Setup) A **Markov Decision Process** is a tuple $(\mathcal{S}, \mathcal{A}, p(y|x,a), r(x), \pi_0)$ where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $p(y|x,a)$ is the transition probability ($y, x \in \mathcal{S}, a \in \mathcal{A}$), $r(x)$ is a reward function, and π_0 is the initial state distribution.*

Given a deterministic horizon T, the learner-interaction is as follows. At the beginning of each episode, an initial state x_0 is chosen according to the state distribution π_0. Afterwards, for each $0 \leq t \leq T$, the learner chooses an action according to a policy $\mu(a_t|x_t)$ where $a_t \in \mathcal{A}, x_t \in \mathcal{S}$. We note that the policy may be a random function. The environment provides a reward $r(x_t)$ and the next state is chosen according to $p(x_{t+1}|x_t, a_t)$. In general, the learner's goal is to maximize the following value function: $J^{\mu} = \mathrm{E}\left[\sum_{t=0}^{T} r(x_t) \middle| x_0 \sim \pi_0, \mu(a|x)\right]$, where the expectation is taken over trajectories with respect to the policy $\mu(a|x)$ and the initial distribution π_0.

When the MDP parameters are given, the problem of finding the policy which maximizes cumulative reward is known in the literature as *planning*. When the MDP parameters are unknown in advance, finding the best policy is known as *Adaptive Control* or *Reinforcement Learning* (RL; Puterman 2005; Bertsekas and Tsitsiklis 1995). The following definition establishes the extended model considered in the paper, denoted by *Contextual MDPs*.

Definition 2 *Contextual Markov Decision Process (CMDP) is a tuple $(\mathcal{C}, \mathcal{S}, \mathcal{A}, \mathcal{M}(c))$ where $\mathcal{C}$ is called the context space, $\mathcal{S}$ and $\mathcal{A}$ are the state and action space, correspondingly, and $\mathcal{M}$ is function mapping any context $c \in \mathcal{C}$ to an MDP $\mathcal{M}(c) = (\mathcal{S}, \mathcal{A}, p^c(y|x,a), r^c(x), \pi_0^c)$.*

So essentially, CMDP is simply a set of models sharing the same state and action space. The simplest scenario in a CMDP setting is when the context is *observable*. In this setting, the problem reduces to correctly generalizing the model from the context. If the observable context c is *finite* where $|\mathcal{C}| = K$, then with no further assumption, one can just learn K different models. In some environments, K scales with the number of sampled trajectories. For instance, consider the problem of targeted advertising: Given behavioral patterns and side information of many customers, companies usually seek to group the consumers so they can target their needs and habits. Since side

information usually resides in a very large set (for example, the cross-product of gender, age, etc.), in practice it is aggregated when the number of clusters depends on the amount of available data.

The model aggregation problem is not considered in this work, and instead we focus on latent contexts for the rest of the paper. Additionally, we assume the initial state distribution and rewards are context independent, maintaining the hardness of the problem while greatly simplifying the writing. Finally, we adopt the common $[0, 1]$-bounded reward assumption.

To differentiate our work from previous models, provided below a short comparison of CMDPs with other known extensions of the MDP model:

1. In **Contextual HMMs** (Wilson and Bobick, 1999; Radenen and Artières, 2014) the context affects only the observation distribution, and there is no control.

2. **POMDPs** (Aberdeen, 2003) are a more complex structure generalizing CMDPs. Since in our case the hidden parameters are constant over time a simpler solution might exist. In addition, some distribution over contexts is assumed.

3. In **Multi-model RL** (Doya et al., 2002) the dynamics and rewards are composed of a convex combination of several models, meaning that in each trajectory there can be more than one valid model.

4. The problem of **state representation** (Maillard et al., 2011) is that of finding a suitable state space for given observations. In our case the state space is the same for all models, allowing more efficient solutions.

5. Models described by **robust MDPs** (Nilim and El Ghaoui, 2005; Wiesemann et al., 2013) consider uncertainty in the transitions and rewards. CMDPs can be viewed as such, where the uncertainty is not rectangular (around each state-action pair) but singular.

Define the general setup as follows: The context space consists of K possible contexts. The time axis is divided into H episodes, denoted by $e_1, \ldots, e_H$. In the beginning of each episode, the environment chooses a context $c \in \mathcal{C}$ (in a random, adversarial or any other fashion). Afterwards, an initial state is randomly chosen according to an initial state distribution π_0^c. Then, for the chosen MDP a length T interaction (where T is a *stopping time*) as described in Definition 1 takes place.

3. Problem Definition and Solution

Assume a small finite $\mathcal{C}$, and that T is bounded almost surely, denoting this setup as *finite sources episodic CMDP*. The goal is maximizing over the cumulative rewards from all trajectories by the H'th trajectory, for increasing H. Therefore, we measured performance with respect to H. Ideally, a good policy should optimize the trade-off between exploration and exploitation of the current chain. However, unlike the standard RL setup, the exploration in this case should consider not only the model's parameters, but also the hidden context. We measure our performance with the notion of *regret*: the difference between the cumulative reward and the cumulative reward obtained by an agent satisfying some optimality property. For example, in infinite horizon RL the cumulative discounted reward is compared against an agent with knowledge of the true model starting from the optimal policy; the faster the regret bound converges to 0 with T the better.

Similarly, we compare ourselves to the all knowing agent applying the optimal policy for the correct context at each trajectory. In our setup, since T is bounded the regret is evaluated mainly

with respect to the number of trajectories H. Notice though, that in each new trajectory some loss is guaranteed until the correct context is identified. Therefore, the regret will always be linear in H. A different optimal agent, when there is some prior distribution over contexts, can be chosen to perform the solution of the resulting POMDP, but there may be other appropriate choices. The problem of redefining the regret to obtain more meaningful bounds was left for future research.

Definition 3 *Define the **regret** for finite sources episodic CMDP problem over H trajectories to be:*

$$Regret = \sum_{h=1}^{H} J_h^* - \sum_{h=1}^{H} \sum_{t=1}^{T_h} r_{h,t} \qquad , \tag{1}$$

where J_h^* is the optimal value function in T_h steps for the context chosen in the h'th trajectory, and $r_{h,t}$ is the reward obtained by the agent in the h'th trajectory at the t'th step.

As a solution, we introduce the CECE general framework (Cluster-Explore-Classify-Exploit) that partitions the trajectories to mini-batches. In the beginning of each mini-batch, all previously seen trajectories are used to form K distinct models via Algorithm 1 (Cluster). Then, for each new trajectory in the current mini-batch the agent generates a partial trajectory using Algorithm 2 (Explore). The partial trajectory is then classified to a context by Algorithm 3 (Classify). Finally, Algorithm 4 sets the policy for the remainder of the trajectory (Exploit). In summary:

Alg. 1: After the previous mini-batch finished, **C**luster observed trajectories to K models. Then, for each trajectory in the current mini-batch:

Alg. 2: **E**xplore the context.

Alg. 3: **C**lassify partial trajectory to model.

Alg. 4: **E**xploit the identified model.

Hence, the algorithm performs online iterative interaction with the environment, and the mini-batches are used to decide on when to take a pause, recluster all seen trajectories, and only then continue. The initial step (before any trajectories were sampled) is contained in the general case, where no trajectories were clustered and so the exploration and exploitation policies can be chosen arbitrarily. Successive mini-batches share all previously seen trajectories, so the clustering gets better.

The following assumptions and theorem guarantee CECE's performance:

Definition 4 *1. Let: $M_i = (\mathcal{S}, \mathcal{A}, p_i(y|x,a), r(x), \pi_0), i = 1, 2$, be two MDPs with the same state space, action space, rewards and initial state distribution. We define M_2 to be an ϵ-**approximate model** of M_1 if for every state-action pair $(s,a) \in \mathcal{S} \times \mathcal{A}$: $\|\Pr_1(\cdot|s,a) - \Pr_2(\cdot|s,a)\|_1 \leq \epsilon$.*

*2. Let: $X_i = (\mathcal{C}_i, \mathcal{S}, \mathcal{A}, \mathcal{M}_i(c)), i = 1, 2$, be two CMDPs with the same state and action space satisfying $|\mathcal{C}_1| = |\mathcal{C}_2|$. We define X_2 to be an ϵ-**approximate CMDP** of X_1 if there exists a matching between the contexts $f : \mathcal{C}_1 \leftrightarrow \mathcal{C}_2$ such that for every $c \in \mathcal{C}_1$ we have that $\mathcal{M}_1(c)$ is an ϵ-approximate model of $\mathcal{M}_2(f(c))$.*

Assumption 1 *Let H_0 be some constant number of trajectories. For every $H > H_0$ there exists $\delta_1(H), \epsilon(H) > 0$, such that after applying Algorithm 1 on H trajectories, with probability at least $1 - \delta_1(H)$ the estimated K-models form an $\epsilon(H)$-approximate CMDP of the true CMDP.*

Assumption 1 guarantees that having enough trajectories will drive Algorithm 1 to output an approximated model for each context. It envelopes a hidden assumption that all contexts were observed enough times. Since there is some probability of error δ_1, the clustering procedure must be repeated when more trajectories are presented to ensure diminishing regret; that is the reason a mini-batch scheme is applied.

Assumption 2 *For every $\epsilon > 0$, there exists $\delta_2(\epsilon)$ such that given an ϵ-approximate CMDP, after applying Algorithms 2 and 3 the correct context is identified with probability at least $1 - \delta_2(\epsilon)$. In addition, the number of steps taken is a stopping time denoted by T_{EC}.*

This assumption assures us each trajectory will be classified correctly with high probability, which will guarantee good performance for exploitation in the next step. Moreover, T_{EC} represents the number of samples needed to differentiate between the models.

Assumption 3 *Given an ϵ-approximate model, Algorithms 4 obtains Regret $\leq \zeta(\epsilon)$.*

Theorem 1 *Let H_i be the number of trajectories in the i'th mini-batch. Then if Assumptions 1, 2, 3 hold, CECE achieves in the L'th mini-batch:*

$$Regret \leq (1-\delta_1)H_L(\delta_2 \mathbb{E}T + (1-\delta_2)(\zeta + \mathbb{E}T_{EC})) + \delta_1 H_L \mathbb{E}T \tag{2}$$

where $\delta_1 = \delta_1(\overline{H}), \epsilon = \epsilon_1(\overline{H}), \delta_2 = \delta_2(\epsilon), \zeta = \zeta(\epsilon)$ and $\overline{H} = \sum_{i=1}^{L-1} H_i$.

Notice that in order for Assumption 1 to hold with a meaningful ϵ, when H_1 is set each model must be observed sufficiently. This fact should be added as an additional assumption depending on the specific realization of Algorithm 1. Supposedly the subsequent H_i's can be chosen arbitrarily small, utilizing information from new trajectories as soon as it is available. Yet, Algorithm 1 may be computationally expensive, making larger H_i's preferable in practice. Another possible approach to this trade-off is to apply on-line clustering (Ailon et al., 2009).

In essence, Algorithm 1 is a form of Multiple Model Learning (MML) algorithm (Vainsencher et al., 2013) – each trajectory is a sample from an unknown model (context) and the goal is learning all models simultaneously. It could also be reduced to the clustering problem, where each trajectory is represented as an $S \times S \times A$ vector of its empirical transition matrix. Indeed, some information is lost in this process: the number of samples from each (s, a) pair in the trajectory is ignored despite its effect on the variance around the sampled distribution. So, ideally each trajectory should be reduced to a point with varying variance across dimensions, which gets smaller for longer trajectories.

Subsequently, one may question whether $\epsilon(H)$ can converge to 0 for infinitely many trajectories. In our setup, as T grows the trajectories are more distinct, but T is bounded almost surely. So even for large T's, there would be at least some constant portion of the trajectories acting as outliers of the model they originated from, possibly tainting the clusters. One way to solve this issue is through an outlier robust clustering (for example K-median).

Next, consider the effect of the trajectories' length T on the hardness of the problem. When T is very large, it is much more important to recognize the correct model. The other extreme case is when T is too small to determine the correct model with high probability. Assuming the models can still be approximated, one reasonable solution would be to try and optimize the worst case performance over all models, an approach closely related to Robust MDPs (Nilim and El Ghaoui,

2005) - a formulation of MDPs with uncertainty in the transitions and rewards (though the specific reduction is intractable; Wiesemann et al. 2013).

When all trajectories are short, it might be impossible to provide an approximation of the true models. Consider for example the extreme case where only one transition is given - unless there is a stationary distribution over contexts the models cannot be learned nor optimized. Subsequently, varied T lengths pose another question: how confident are we in the clustering of each trajectory? Embedding short trajectories might inject more noise to the clustering process than improve it, so some selection is needed to insure proper modeling. This question may relate to the notion of clusters separability (Ostrovsky et al., 2006) - short trajectories can lead to non-separable models that cannot be learned through clustering.

A rather simple realization of Algorithm 2 (exploration) is to apply a fixed policy until some condition is fulfilled. One may consider what is the policy which will achieve this condition with as few steps as possible (since the regret is linear in the number of exploration steps).

3.1 A Specific Instance

We give an example for an instance of CECE and substitute in Assumptions 1, 2, 3. For simplicity, we assume the trajectory length is a constant T for the remainder of the analysis. The proposed realization was chosen to be trivial to allow simple analysis; It is only a demonstration of the trade-offs in CMDPs and CECE's modularity. Algorithm 1 is the following scheme:

1. For each trajectory h, and state action pair (s, a), estimate the transition probability $\widehat{\Pr}_h(\cdot|s, a)$ by its empirical distribution.

2. Go over all possible partitions of trajectories to K sets $\{C_k\}_{k=1}^K$, and minimize over the following score:
$$\sum_{k=1}^{K} \sum_{h \in C_k} \max_{s,a} \|\widehat{\Pr}_h(\cdot|s, a) - \widehat{\Pr}_k(\cdot|s, a)\|_1, \tag{3}$$
where $\widehat{\Pr}_k(\cdot|s, a)$ is the estimated transition probability for all trajectories in the cluster.

This scheme is highly inefficient as it performs an exhaustive search for the best partition. However, as a preliminary result all we require is for it to accommodate Assumption 1. There are other polynomial time clustering algorithms with guarantees (Ostrovsky et al. 2006; Arthur and Vassilvitskii 2007 for instance), but their bounds and assumptions would have to be adjusted to our case. In Algorithm 2, the uniform policy over actions is applied for a constant number of steps T_{EC}. The proposed Algorithm 3 chooses the model obtaining the smallest L_1 distance between the set of models and the empirical transition matrix from the partial trajectory. Lastly, Algorithm 4 was chosen naively to apply the exploitation policy with regards to the estimated model.

Assumption 4 *Let $\alpha, \beta \in (0, 1)$.*

1. *By the H'th trajectory, each model was sampled at least βH times.*

2. *For some D, for every two contexts c_1, c_2 and s, a: $\|\Pr_{c_1}(\cdot|s, a) - \Pr_{c_2}(\cdot|s, a)\| \geq D$.*

3. *In every trajectory, each state-action pair is visited at least αT times, and T is large enough: $T \in O(\frac{S}{\alpha D^2} \log(\frac{KSA}{D}))$.*

Lemma 1 *Under Assumption 4, the described realization satisfies Assumptions 1-3 with:*

$$\begin{aligned} &\epsilon(H) \in O(KSAe^{S-\alpha TD^2}), \quad \zeta(\epsilon) \in O(S^2T^2\epsilon) \\ &\delta_1(H) \in O(KSAe^{S-\alpha T\beta HD^2}), \quad \delta_2(\epsilon) \in O(Ke^{S-T_{EC}(\frac{D}{2}-\epsilon)^2}), \quad D > 2\epsilon. \end{aligned} \tag{4}$$

The full proof is available in Section C of the supplementary material.

Corollary 1 *Under Assumption 4, the described realization achieves in the L'th mini-batch:*

$$Regret \leq O(H_L TKe^{S-T_{EC}D^2/4} + (H_L T^2 KS^3 Ae^{S-\alpha TD^2} + H_L T_{EC}) + H_L TKSAe^{S-\alpha T\beta \overline{H} D^2}), \tag{5}$$

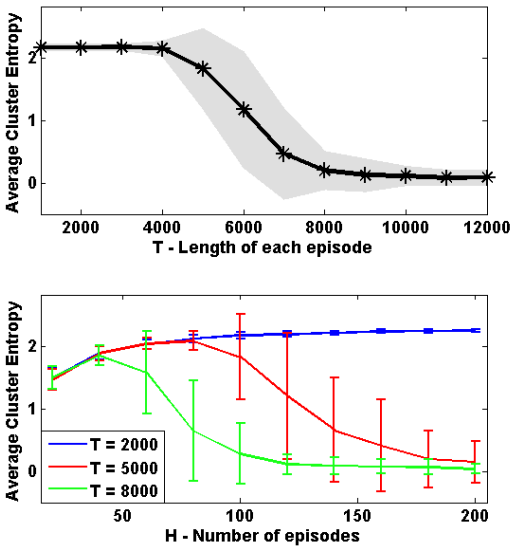
where $\overline{H} = \sum_{i=1}^{L-1} H_i$.

Notice that each summand relates to a different error: The first summand corresponds to trajectory misclassification - it can point us to proper choice of T_{EC}: scaled with S and the distance between models. The second, corresponds to the context and model uncertainty - large T and α are required to estimate each model well enough. The third corresponds to trajectories misclustering - it is the only error which diminishes with H, as the exponential multiplicative converges to 0.

4. Experiments

In this section we discuss the trade-offs that exist in the CMDPs settings. In the first experiment we test only the clustering part in CECE. We consider a CMDP with $K = 5$ equal probability contexts, $|\mathcal{A}| = 2$ actions and $|\mathcal{S}| = 100$ states where the transition matrix for each context was drawn from a uniform distribution. We generate H trajectories of a constant length T sampling actions uniformly. For the purpose of scoring the clusters we calculate the entropy of each distribution over clusters for each correct context, and average the results according to the number of samples from that context. Thus, when the trajectories are perfectly clustered, for each context the entropy will be 0 and so will be the average. The worst possible score $\log(K)$ results from independent clusters and contexts. The clustering algorithm we used in this case was K-means on the vectorized empirical transition matrices, the results were averaged over 100 trials and were added error bars of one standard deviation. In the first part of the experiment (top plot in Figure 1) we generate $H = 100$ trajectories and present the score as a function of the trajectories length T. In the second part of the experiment (bottom plot in Figure 1), we generate trajectories of varying lengths $T = 2000, 5000, 8000$ and measure the score as a function of the number of episodes H.

Figure 1: Experiment 1



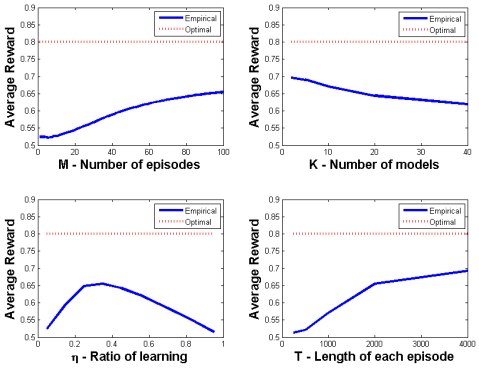
We conclude the following: (1) There is a phase transition in the clustering performance with respect to T: below a certain threshold (here $T = 4000$) the clustering utterly fails, followed by a short adjustment period, where finally (here at $T = 8000$) the clustering succeeds almost certainly. (2) If the trajectories are too short, the clustering will fail even when increasing the number of episodes. (3) If the trajectories are sufficiently long, additional episodes improve the clustering quality (as implied by Lemma 1).

Next, we experimented with the full CECE algorithm. We simulated a CMDP with $|\mathcal{S}| = 100$ states, $|\mathcal{A}| = 4$ actions and $K = 20$ contexts of equal probability. Each trial consists of $H = 100$ episodes of length $T = 2000$. The results were averaged over 20 experiments. The parameter T_{EC} sets the portion of the trajectory time steps dedicated to identify the model, and was taken to be $\eta \cdot T$, $\eta = 0.3$. The learning policy employed by Algorithm 2 was taken to be uniform over all actions. The exploitation algorithm used is Q-learning Bertsekas and Tsitsiklis (1995).

We performed four experiments where in each of the experiments all the parameters excluding one were fixed (when K changes the number of clusters changes accordingly). The average reward throughout the experiment is measured. The results are presented in Figure 2. On the top-left and bottom-right plots we can see how CECE behaves as the number of episodes and trajectory length increase. As more data are available, the average reward increases since the clustering phase performs better and the models are better learned. Similarly, the average reward decreases as more models are introduced (top-right plot) since it is harder to cluster and learn each model. Notice that for constant proportion $\frac{T_{EC}}{T}$ there will always be a difference between the optimal and the achieved value due to the identification phase.

An interesting result is presented in the bottom-left plot. The parameter $\eta = \frac{T_{EC}}{T}$ describing the portion of samples taken to identify the correct model. The resulting plot represents the exploration-exploitation trade-off for our suggested model: How many samples are used to identify the correct model against how many of them are used to optimize the C-MDP.

Figure 2: Experiment 2



5. Conclusions and Future Work

In this work we presented a new framework for modeling multiple Markovian sources with sequential decision making. While our models can be encompassed in existing models (e.g., POMDPs; Aberdeen 2003) the proposed setup offers much flexibility in modeling both observable and latent static context while maintaining computational tractability. We demonstrated that under certain conditions one can overcome two fundamental problems: (1) learning the model parameters, and (2) optimizing on-line the action within an RL framework. We suggested and analyzed basic algorithms when the number of contexts is finite. However, from an algorithmic and analytic points of view the theoretical trade-off between learning, exploration, optimization, and control of CMDPs is still very much an open question.

References

Douglas Aberdeen. A (revised) survey of approximate methods for solving partially observable markov decision processes. *National ICT Australia, Canberra, Australia, Tech. Rep*, 2003.

Nir Ailon, Ragesh Jaiswal, and Claire Monteleoni. Streaming k-means approximation. In *Advances in Neural Information Processing Systems*, pages 10–18, 2009.

David Arthur and Sergei Vassilvitskii. k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1027–1035. Society for Industrial and Applied Mathematics, 2007.

Dimitri P Bertsekas and John N Tsitsiklis. Neuro-dynamic programming. Athena Scientific, 1995.

K. Doya, K. Samejima, K. Katagiri, and M. Kawato. Multiple model-based reinforcement learning. *Neural computation*, 14(6):1347–1369, 2002.

Michael Kearns and Satinder Singh. Near-optimal reinforcement learning in polynomial time. *Machine Learning*, 49(2-3):209–232, 2002.

Odalric-Ambrym Maillard, Daniil Ryabko, and Rémi Munos. Selecting the state-representation in reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 2627–2635, 2011.

Arnab Nilim and Laurent El Ghaoui. Robust control of markov decision processes with uncertain transition matrices. *Operations Research*, 53(5):780–798, 2005.

Rafail Ostrovsky, Yuval Rabani, Leonard J Schulman, and Chaitanya Swamy. The effectiveness of lloyd-type methods for the k-means problem. In *Foundations of Computer Science, 2006. FOCS'06. 47th Annual IEEE Symposium on*, pages 165–176. IEEE, 2006.

Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2005.

Mathieu Radenen and Thierry Artières. Handling signal variability with contextual markovian models. *Pattern Recognition Letters*, 35:236–245, 2014.

Daniel Vainsencher, Shie Mannor, and Huan Xu. Learning multiple models via regularized weighting. In *Advances in Neural Information Processing Systems*, pages 1977–1985, 2013.

Tsachy Weissman, Erik Ordentlich, Gadiel Seroussi, Sergio Verdu, and Marcelo J Weinberger. Inequalities for the l1 deviation of the empirical distribution. *Hewlett-Packard Labs, Tech. Rep*, 2003.

Wolfram Wiesemann, Daniel Kuhn, and Berç Rustem. Robust markov decision processes. *Mathematics of Operations Research*, 38(1):153–183, 2013.

Andrew D Wilson and Aaron F Bobick. Parametric hidden markov models for gesture recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 21(9):884–900, 1999.

Appendix A. List of Notations

Notation	Meaning
$\mathcal{S}$	State space or number of states
$\mathcal{A}$	Action space or number of actions
T	Time horizon
t	Time index $t = 0..T$
H	Number of trajectories in batch data
H_L	Number of trajectories in the L'th mini-batch
$\mathcal{C}$	Number of possible contexts
J^{μ}_{M}	Value of policy μ in model M
D	Minimal inf-distance between two distinct models.

Appendix B. Useful Lemmas

The following Lemmas are used in the proofs:

Lemma 2 *Weissman et al. (2003) Let P be a probability distribution on the set $\mathcal{S} = 1, .., S$. Let $\mathbb{X}^m = X_1, X_2, ..., X_m$ be independent identically distributed random variables distributed according to P. Then for all $\epsilon > 0$,*

$$\Pr(\|P - \hat{P}_{\mathbb{X}^m}\|_1 \geq \epsilon) \leq e^{S - m\epsilon^2/2} \tag{6}$$

Lemma 3 *Kearns and Singh (2002) Let M be an MDP over S states, and $\hat{M}$ be an $O(\epsilon)$-approximation of M. Then for any policy μ:*

$$|J^{\mu}_{M} - J^{\mu}_{\hat{M}}| \leq S^2T^2\epsilon, \tag{7}$$

and consequently for the optimal policy in each MDP correspondingly:

$$|J^{*}_{M} - J^{*}_{\hat{M}}| \leq 3S^2T^2\epsilon, \tag{8}$$

Appendix C. Proof of Lemma 1

Lemma 1 *If Assumption 4 holds, the described realization of Algorithms 1-4 satisfy Assumptions 1-3 with:*

$$\begin{aligned} &\epsilon(H) \in O(KSAe^{S-\alpha TD^2}), \\ &\delta_1(H) \in O(KSAe^{S-\alpha T\beta HD^2}), \\ &\delta_2(\epsilon) \in O(Ke^{S-T_{EC}(\frac{D}{2}-\epsilon)^2}), \quad D > 2\epsilon \\ &\zeta(\epsilon) \in O(S^2T^2\epsilon). \end{aligned} \tag{9}$$

Proof We show each Assumption holds, starting with Assumption 1.

For two transition functions P_1, P_2 of size $S \times S \times A$ denote:

$$\|P_1 - P_2\| \triangleq \max_{s,a} \|P_1(\cdot|s,a) - P_2(\cdot|s,a)\|. \tag{10}$$

We denote by $\widehat{\Pr}_h, \widehat{\Pr}_c$ the estimated transition matrices from trajectory h and cluster c correspondingly. In addition, C^* is the true clustering of each trajectory, and C^{opt} is the clustering found by the algorithm.

Since there are at least αT samples from each state-action pair, according to Lemma 2 and the union bound, we obtain that:

$$\Pr(\|\widehat{\Pr}_h(\cdot|s,a) - \Pr_{C^*(h)}(\cdot|s,a)\| \leq \epsilon) \geq 1 - SAe^{S-\alpha T\epsilon^2/2}. \tag{11}$$

Since there are at least βH trajectories from each model, we also obtain that:

$$\Pr(\|\widehat{\Pr}_{C^*(h)}(\cdot|s,a) - \Pr_{C^*(h)}(\cdot|s,a)\| \leq \epsilon) \geq 1 - SAe^{S-\alpha T\beta H\epsilon^2/2}, \tag{12}$$

and therefore:

$$\Pr(\sum_{h=1}^{H} \|\widehat{\Pr}_{C^*(h)}(\cdot|s,a) - \Pr_{C^*(h)}(\cdot|s,a)\| \leq H\epsilon) \geq 1 - KSAe^{S-\alpha T\beta H\epsilon^2/2}, \tag{13}$$

Now we obtain the following:

$$\begin{aligned}
\sum_{h=1}^{H} \|P_{C^*(h)} - \hat{P}_{C^{opt}(h)}\| &\leq \sum_{h=1}^{H} \|P_h - P_{C^*(h)}\| + \sum_{h=1}^{H} \|P_h - \hat{P}_{C^{opt}(h)}\|, \quad \textit{Triangle inequality} \\
&\leq \sum_{h=1}^{H} \|P_h - P_{C^*(h)}\| + \sum_{h=1}^{H} \|P_h - \hat{P}_{C^*(h)}\|, \quad \textit{By Algorithm definition} \\
&\leq 2\sum_{h=1}^{H} \|P_h - P_{C^*(h)}\| + \sum_{h=1}^{H} \|P_{C^*(h)} - \hat{P}_{C^*(h)}\|, \quad \textit{Triangle inequality (second term).}
\end{aligned} \tag{14}$$

When H is large, we can approximate $2\sum_{h=1}^{H} \|P_h - P_{C^*(h)}\| \in O(H(1-\delta)\epsilon + H\delta) = O(H\epsilon + H\delta)$ for $\delta = SAe^{S-\alpha T\epsilon^2/2}$ since each summand is bounded by ϵ with that probability, and when it is unbounded the maximal value of L_1 distance between two distributions is a constant 2. Therefore:

$$\frac{1}{H}\sum_{h=1}^{H} \|P_{C^*(h)} - \hat{P}_{C^{opt}(h)}\| \in O(\epsilon + \delta) \tag{15}$$

with probability at least $1 - KSAe^{S-\alpha T\beta H\epsilon^2/2}$, for $\delta = SAe^{S-\alpha T\epsilon^2/2}$.

Since the average is of that order, there must exist a matching between the true clusters and optimal clusters satisfying:

$$\max_{c \in C*} \|P_c - \hat{P}_{c^{opt}}\| \in O(\epsilon + \delta) \tag{16}$$

If the distance between every two true clusters is $D > O(\epsilon + \delta)$, the agreement between matching clusters are on all trajectories in a reasonable radius, i.e. $O(1-\delta)$ of the trajectories. so the error in each model is of the order $O(K\delta)$:

$$\|\Pr_c(\cdot|s,a) - \widehat{\Pr}_i(\cdot|s,a)\| \leq KSAe^{S-\alpha T\epsilon^2/2}. \tag{17}$$

Now in order for $D > O(\epsilon + \delta)$ to hold, we can choose ϵ to be of order D, and then:

$$KSAe^{S-\alpha T\epsilon^2/2} \in O(D) \Rightarrow T \in O(\frac{S}{\alpha D^2}\log(\frac{KSA}{D})). \tag{18}$$

To summarize, for $T \in O(\frac{S}{\alpha D^2}\log(\frac{KSA}{D}))$ we obtain that with probability at least $1-\delta(H)$, $\|\Pr_c(\cdot|s,a) - \widehat{\Pr}_i(\cdot|s,a)\| \leq \epsilon(H)$, where:

$$\epsilon(H) \in O(KSAe^{S-\alpha TD^2}), \qquad \delta(H) \in O(KSAe^{S-\alpha T\beta HD^2}). \tag{19}$$

Next, we show Assumption 2 holds. We bound the probability of misclassification by the following probability:

$$\Pr(\|\widehat{P}_h - \widehat{P}_{C(h)}\| \leq \frac{D}{2}, \|\widehat{P}_h - \widehat{P}_{c\neq C(h)}\| \geq \frac{D}{2}), \tag{20}$$

as if this event occurs then the true model will be chosen. To bound this quantity, we use the union bound over the complement event, so we need to bound:

$$\Pr(\|\widehat{P}_h - \widehat{P}_{C(h)}\| \geq \frac{D}{2}), \qquad \Pr(\|\widehat{P}_h - \widehat{P}_{c\neq C(h)}\| \leq \frac{D}{2}). \tag{21}$$

For the left term:

$$\begin{aligned}\Pr(\|\widehat{P}_h - \widehat{P}_{C(h)}\| \geq \frac{D}{2}) &\leq \Pr(\|\widehat{P}_h - P_{C(h)}\| + \|P_{C(h)} - \widehat{P}_{C(h)}\| \geq \frac{D}{2}) \\ &\leq \Pr(\|\widehat{P}_h - P_{C(h)}\| \geq \frac{D}{2} - \epsilon) \\ &\leq e^{S-T_{EC}(\frac{D}{2}-\epsilon)^2/2}, \qquad \textit{Lemma 2}\end{aligned} \tag{22}$$

For the right term:

$$\begin{aligned}\Pr(\|\widehat{P}_h - \widehat{P}_{c\neq C(h)}\| \leq \frac{D}{2}) &\leq \Pr(\|\widehat{P}_h - P_{C(h)}\| - \|P_{c\neq C(h)} - P_{C(h)}\| - \|P_{c\neq C(h)} - \widehat{P}_{c\neq C(h)}\| \leq \frac{D}{2}) \\ &\leq \Pr(\|\widehat{P}_h - P_{C(h)}\| \leq \frac{D}{2} + \epsilon + D) \\ &\leq e^{S-T_{EC}(\frac{3D}{2}+\epsilon)^2/2}, \qquad \textit{Lemma 2}\end{aligned} \tag{23}$$

Now, using the union bound we obtain that the classification is correct with probability at least $1-\delta$, where

$$\delta = e^{S-T_{EC}(\frac{D}{2}-\epsilon)^2/2} + Ke^{S-T_{EC}(\frac{3D}{2}+\epsilon)^2/2} \tag{24}$$

■