

Cost-Bounded Active Classification Using Partially Observable Markov Decision Processes

Bo Wu, Mohamadreza Ahmadi, Suda Bharadwaj, and Ufuk Topcu

Abstract—Active classification, i.e., the sequential decision making process aimed at data acquisition for classification purposes, arises naturally in many applications, including medical diagnosis, intrusion detection, and object tracking. In this work, we study the problem of actively classifying dynamical systems with a finite set of Markov decision process (MDP) models. We are interested in finding strategies that actively interact with the dynamical system, and observe its reactions so that the true model is determined efficiently with high confidence. To this end, we present a decision-theoretic framework based on partially observable Markov decision processes (POMDPs). The proposed framework relies on assigning a classification belief (a probability distribution) to each candidate MDP model. Given an initial belief, some misclassification probabilities, a cost bound, and a finite time horizon, we design POMDP strategies leading to classification decisions. We present two different approaches to find such strategies. The first approach computes the optimal strategy “exactly” using value iteration. To overcome the computational complexity of finding exact solutions, the second approach is based on adaptive sampling to approximate the optimal probability of reaching a classification decision. We illustrate the proposed methodology using two examples from medical diagnosis and intruder detection.

I. INTRODUCTION

We consider the following scenario as a running example. A doctor needs to implement certain tests and treatments to determine which disease a patient has in order to prescribe further specialized tests and/or treatments. The doctor makes a diagnosis from a finite set of (known) candidate diseases, which can evolve dynamically, e.g. the symptoms can change, with time, tests, and treatment choices. Each test or treatment, which is implemented sequentially, can incur uncertain reactions and some costs to the patient. For each candidate disease, it is possible to probabilistically predict the patient’s reaction to tests or treatments based on available historical data. At each step, the doctor is required to decide whether to make a diagnosis or to implement further tests to gather more information based on the history of the patient’s test results as well as the predicted reactions from the patients. On the other hand, the diagnosis procedure should not last too long, since the patient’s health condition may worsen. Additionally, the cost to reach a diagnosis should be bounded to make it affordable to the patient. Such a classification procedure requires sequentially planning of the tests, while respecting a given cost.

Bo Wu, Mohamadreza Ahmadi, Suda Bharadwaj and Ufuk Topcu are with the Department of Aerospace Engineering and Engineering Mechanics, and the Institute for Computational Engineering and Sciences (ICES), University of Texas, Austin, 201 E 24th St, Austin, TX 78712. email: {bwu3, mrahmadi, suda.b, utopcu}@utexas.edu

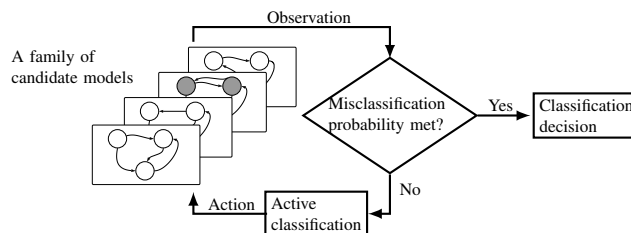


Fig. 1: Active classification from a family of candidate models. The model with shaded states is the true one.

The conventional approach for classification, which we refer to as *passive* classification, is an open-loop process that determines the class of the object of interest based on whatever data provided. While a challenging problem itself, such an approach is not suitable for the scenario described above. Passive classification cannot decide what, when and how to obtain data to best assist the classification task. As a result, excessive amount of data (for example, the tests in diagnosis scenario) may have to be collected. Furthermore, potentially uninformative data and ignorance of the impact of the data collection process on the object of interest may make the classification results less accurate.

Active classification [1], on the other hand, is a sequential decision-making process that has control over the data acquisition and interacts closely with the object of interest. Compared to passive classification, active classification is a more plausible approach in many practical classification applications, such as medical diagnosis [2], [3], intrusion detection, and target tracking [4], where the object of interest has certain underlying dynamics belonging to a family of models. Each model is determined from the history data. The classification process directly interacts with the state evolution of the object by selecting actions and obtaining observations as a closed-loop system such that classification decisions can be reached more efficiently and accurately (see Figure 1).

In this paper, in order to capture the stochastic uncertainties of the outcomes associated with each action (such as a treatment) during classification, we assume the underlying dynamic model of the object to be classified belongs to a family of Markov decision processes (MDPs). Then, we formalize an active classification problem that can be cast into the framework of partially observable Markov decision processes (POMDPs).

As a comprehensive model for planning in partially observable stochastic environments, POMDPs characterize the uncertainties in the state evolution due to actuation imperfec-

tions or interactions with a stochastic environment as well as lack of environmental information or the observation noises. POMDPs have been applied to a variety of applications in medical diagnosis [2], [3], health care [5], robotic sensing and perception [1], [6], [7], [8] and wireless communication [9]. Most existing results, however, are based on minimizing the costs incurred by actions and classification uncertainty as measured in terms of entropy [10], where the classification accuracy is indirectly embedded in the rewards.

Because the underlying true model is not known, it is only possible to maintain a belief, which in the proposed setting, is a probability distribution over the possible MDP models and evolves based on the history of observations and actions. The classification decision is made whenever the probability of a certain MDP model being the true model of the system exceeds a given threshold based on the misclassification probability, as shown in Figure 1. Such a framework is particularly suitable to time- and cost-bounded classification tasks. The classification is translated to a sequential decision-making problem under uncertainties. Our objective is to find a strategy to dynamically select classification actions such that with an optimized probability, a classification decision can be made with a bounded cost. A similar problem was studied recently in [11], where the belief in a POMDP must reach some goal states. However, in [11], such requirement must be satisfied with probability 1, which could be restrictive, considering the probabilistic uncertainties in the belief dynamics.

Given the POMDP model and the desired misclassification probability, we propose two approaches to obtain the classification strategy that optimizes the probability to reach a classification decision. The exact solution is inspired by the cost bounded reachability in MDPs [12]. It relies on obtaining the underlying MDP whose states are beliefs from the original POMDP. This recursive process terminates once the cost bound, a classification decision, or the maximum number of steps is reached. Then the optimal strategy can be computed from the obtained belief MDP. To overcome the computational complexity of finding exact solutions for POMDPs, our second approach adaptively samples the actions in the belief MDP to approximate the optimal probability to reach a classification decision.

The rest of this paper is structured as follows. Section II provides the necessary modeling preliminaries. Section III formulates the problem. Two algorithms are presented in Section IV to find the optimal strategy to reach a classification decision with the bounded cost. Simulation results are presented in Section V. Section VI concludes the paper.

II. PRELIMINARIES

In this section, we describe preliminary notions and definitions used in the sequel.

A. Markov Decision Processes [13]

Formally, a Markov decision process (MDP) is defined as follows.

Definition An MDP is a tuple $\mathcal{M} = (S, \hat{s}, A, T, C)$ where

- S is a finite set of states;
- \hat{s} is the initial state;
- A is a finite set of actions;
- $T : S \times A \times S \rightarrow [0, 1]$ is the probabilistic transition function with $T(s, a, s') := P(s_t = s' | s_{t-1} = s, a_{t-1} = a), \forall s, s' \in S, a \in A$; and
- $C : S \times A \rightarrow \mathbb{R}_{\geq 0}$ is the cost function.

In an MDP, a finite state-action path is $\omega = s_0 a_0 s_1 a_1 \dots$, where $s_i \in S, a_i \in A$ and $T(s_i, a, s_{i+1}) > 0$.

B. Hidden-Model MDPs [14]

The classification problem assumes that the object of interest is an unknown MDP that belongs to a known finite set of MDPs. Formally, it can be modeled as a hidden model MDP (HMMDP) where the underlying true MDP is known to be one out of a finite set $\mathcal{M} = \{\mathcal{M}_i | i \in \mathcal{C}\}$, $\mathcal{C} = \{1, \dots, L\}$, i.e., there are L candidate models in total. We assume, without loss of generality, the MDPs share the same state space S , initial state \hat{s} , the action set A and cost function C .

Given the initial state \hat{s} , we denote $\hat{b}_{\hat{s}}(i)$ as the initial probability that \mathcal{M}_i is the underlying true model. Then an HMMDP is essentially a partially observable Markov decision process (POMDP) $\mathcal{P} = (Q, \pi, A, T, Z, O, C)$ where

- $Q = S \times \mathcal{C}$;
- $\pi : Q \rightarrow [0, 1]$ is the initial state distribution with $\pi(s, i) = \hat{b}_{\hat{s}}(i)$, if $s = \hat{s}$ and 0 otherwise;
- A is a finite set of actions;
- $T((s, i), a, (s', i')) = \begin{cases} T_i(s, a, s') & \text{if } i = i' \\ 0 & \text{otherwise;} \end{cases}$
- $Z = S$ is the set of all possible observations;
- $O : Q \times Z \rightarrow [0, 1]$ is the observation function with $O((s, i), z) = 1$ if $z = s$ and 0 otherwise;
- $C : S \times A \rightarrow \mathbb{R}_{\geq 0}$ is the cost function.

The definition of T implies that the underlying true model \mathcal{M}_i will not change to any other model during the classification process. From the definition of the observation function O , it can be seen that the observation gives the perfect information about the state element s in the state-model tuple (s, i) , but not the model element i . Therefore, when s is observed, we denote $b(s, i) = b_s(i) \geq 0$ for simplicity.

For classification purposes, it is essential to keep track of a belief b where $b(i) \in [0, 1], \sum_{i \in \mathcal{C}} b(i) = 1$, which is a probability distribution over all the possible MDP models. The belief space $B \subseteq \mathcal{R}^L$ represents the set of all possible beliefs. It is then possible to obtain a belief MDP $\mathcal{B} = (B, \hat{b}_{\hat{s}}, A, T, C_B)$ where

- $b \in B$ is the belief;
- the belief state transition probability is described by

$$T(b'_{s'}, a, b_s) = \sum_i b_s(i) T_i(s, a, s'), \quad (1)$$

with

$$b'_{s'}(i) = \frac{T_i(s, a, s') b_s(i)}{\sum_j T_j(s, a, s') b_s(j)}; \text{ and} \quad (2)$$

- $C_B(b_s, a) = C(s, a)$ is the cost associate to executing action a at state s .

III. PROBLEM FORMULATION

To make a classification decision within a finite time bound H , we keep track of the belief b_s and claim the underlying model belongs to \mathcal{M}_i , whenever

$$b_s(i) \geq \lambda_i, \quad (3)$$

where $\lambda^i \in (0.5, 1]$ denotes the minimum confidence to claim that the system belongs to \mathcal{M}_i . Equivalently, $1 - \lambda^i$ represents the maximal acceptable error rate for \mathcal{M}_i . Such a belief state is a terminal state in the belief MDP. Once the terminal state is reached, the classification task is accomplished. To make the classification decision unique, i.e., there will be only one model to be declared true according to (3), we require that $\lambda_i > 0.5, i \in \mathcal{C}$. We denote $G = \cup_{i \in \mathcal{C}} G_i$ where $G_i = \{b | b_s(i) \geq \lambda_i\}$ as the set of beliefs that should be reached to make a classification decision.

In addition to the reachability requirement, it is also essential to accomplish the classification task with a fixed amount of cost. That is, for state-action path ω in the belief MDP \mathcal{B} where $\omega = b_0 a_0 b_1 a_1 \dots b_N$ such that $b_i \notin G, i < H$ and $b_N \in G, N \leq H$, it is required that

$$C(\omega) \leq D, \text{ where } C(\omega) = \sum_{i=0}^{N-1} C(a_i). \quad (4)$$

Here ω denotes a path that a classification decision is met for the first time, where $C(\omega)$ represents its accumulated cost. We denote Ω_G as the set of such paths that reach G within time bounds H and cost bound D .

For the classification task, the objective is to compute a policy μ to dynamically select classification actions. Since the classification should have bounded horizon and costs, the policy μ is $\mu = \{\mu_i | \mu_i : B \times E \rightarrow A, 0 \leq i \leq H\}$, which maps the current time step $i \in [0, H]$, the belief state $b \in B$ and accumulated cost $e \in E = \mathbb{N}$, to an action $a \in A$.

Given a strategy μ to resolve the nondeterminism in the action selection in the HMMDP model and a path $\omega = b_0 a_0 b_1 a_1 \dots b_N$, it is possible to calculate its probability

$$P(\omega) = \prod_i T(b_i, a_i, b_{i+1}),$$

where $\mu_i(b_i, c_i) = a_i, c_i = \sum_{j=0}^i C(a_j)$. Thus, with a strategy μ , it is possible to compute the probability $P_\mu^{\leq D}(\diamond^{\leq H} G)$ to reach a classification decision within time bound H and cost bound D , where

$$P_\mu^{\leq D}(\diamond^{\leq H} G) = \sum_{\omega \in \Omega_G} P(\omega).$$

All in all, our objective is to compute a strategy μ^* such that

$$\mu^* = \operatorname{argmax}_\mu P_\mu^{\leq D}(\diamond^{\leq H} G), \quad (5)$$

i.e., μ^* achieves the maximum probability to reach the decision region G in H steps with the accumulated cost no larger than D .

In many cases, reaching a classification decision may not be the only objective. For example, it is more desired to reach a diagnosis decision at the early or intermediate stages of a disease. For robotics applications, it is more desirable to classify the object of interest without colliding with it. Therefore, the belief may have to be constrained within a set of *safe* belief $B_{safe} \subseteq B$ before the classification decision is reached. In such cases, the set of goal states G is defined to be $G = \cup_{i \in \mathcal{C}} G_i$ where $G_i = \{b | b \in B_{safe}, b_s(i) \geq \lambda_i\}$ and the objective is to find a policy μ^* such that

$$\mu^* = \operatorname{argmax}_\mu P_\mu^{\leq D}(B_{safe} U^{\leq H} G), \quad (6)$$

which denotes that maximized probability for the belief state to reach a classification decision G while remaining in B_{safe} , considering the bounds on the horizon H and cost D . It can be seen that (5) is a special case of (6) where $B_{safe} = B$.

IV. COST-BOUNDED ACTIVE CLASSIFICATION

Given an HMMDP model with the corresponding MDPs $\mathcal{M}_i, i \in \mathcal{C}$, the time bound H and cost bound D , in this section we introduce two approaches to solve the active classification problem as defined by (5). Then we discuss how the solutions generalize to (6).

A. Exact Solution

Since the classification decision is defined on belief states as shown in (3), the first step is to obtain a finite belief-state MDP \mathcal{B} from the HMMDP model considering the accumulated cost. Such a procedure is called *unfolding* and inspired by the similar treatment in MDPs [12] for cost-bounded properties. In this paper, we extend this procedure to POMDPs. Without considering the reachability and cost constraints as defined in (3) and (4), the approach to obtain \mathcal{B} is in Section II-B. However, with cost constraints, the state space Q of the belief MDP \mathcal{B} is the product of the belief B and the accumulated cost E .

The algorithm for obtaining \mathcal{B} is shown in Algorithm 1. It is essentially a recursive breadth-first traversal starting from the initial state $\hat{q} = (\hat{b}_s, 0)$ with initial belief \hat{b}_s and accumulated cost of 0. The algorithm goes on for H iterations or until there is no more state to be expanded, as can be seen in Line (2). At each iteration i , we iterate through every state $q = (b, e)$ to be expanded (Line 4), where b is the belief state, e is the cost accumulated so far. For each action $a \in A$ (Line 5), we calculate its next accumulated reward e' (Line 6). We terminate the expanding if $e' > D$ (Line 7), i.e., when the cost bound is exceeded. Otherwise, the successor belief state b' is computed (Line 9) as well as the transition probability (Line 10). If $b' \in G$, a classification decision is reached, otherwise $q' = (b', e')$ is added to a set *Next* and will be expanded in the next iteration (Line 14). By construction, it is not hard to see that no state in \mathcal{B} will violate the cost bound and every state whose belief component belongs to G is a terminal state.

From the output of Algorithm 1, it can be observed that the accumulated cost is already encoded in the state space of \mathcal{B} , therefore the cost function component in \mathcal{B} is omitted.

Algorithm 1: Cost-Bounded Unfolding

input : An HMDP model with MDPs $\mathcal{M}_i, i \in \mathcal{C}$, $\mathcal{C} = \{1, \dots, L\}$, time bound H , cost found D , initial belief $b_{\hat{s}}$ and reachability constraint as defined in (3).

output: Finite state MDP $\mathcal{B} = (Q = B \times E, \hat{b}, A, T)$.

```
1  $Q = \{(\hat{b}_{\hat{s}}, 0)\}, Cur = \{(\hat{b}_{\hat{s}}, 0)\}, i = 0;$ 
2 while  $i < H$  and  $Cur \neq \emptyset$  do
3    $Next = \{\};$ 
4   for  $q = (b_s, e) \in Cur$  do
5     for  $a \in A$  do
6        $e' = e + C(s, a);$ 
7       if  $e' \leq D$  then
8         for  $s' \in S$  do
9           Compute  $b'$  according to (2);
10          Let  $q' = (b', e')$ ,  $T(q, a, q')$  is
              computed according to (1);
11          if  $q' \notin Q$  then
12             $Q = Q \cup q';$ 
13            if  $b' \notin G$  then
14               $Next = Next \cup q';$ 
              end
            end
          end
        end
      end
    end
   $Cur = Next;$ 
end
return  $\mathcal{B};$ 
```

Once \mathcal{B} is obtained, it is then possible to calculate the optimal strategy μ^* on \mathcal{B} to achieve the following probability

$$P_{max}(\diamond^{\leq H} G) = P_{\mu^*}(\diamond^{\leq H} G), \quad (7)$$

i.e., the maximized probability to reach a classification decision within H steps but without considering the cost bound.

To get μ^* , it is needed to compute the maximal probability, denoted as $P_{max}^q(\diamond^{\leq k} G)$ to reach G with in $k \in \{0, \dots, H\}$ steps from any $q \in Q$:

$$P_{max}^q(\diamond^{\leq H} G) = P_{max}(\diamond^{\leq H} G).$$

We first divide the state set Q into two disjoint subsets $Q^{yes} = \{q = (b, e) | b \in G\}$ and $Q^? = Q \setminus Q^{yes}$. The computation of $P_{max}^q(\diamond^{\leq k} G)$ is essentially a dynamic program as shown below.

$$P_{max}^q(\diamond^{\leq i} G) = 1, \quad \forall q \in Q^{yes}, i \in \{0, \dots, H\}, \quad (8)$$

$$P_{max}^q(\diamond^{\leq 0} G) = 0, \quad \forall q \in Q^?, \text{ and} \quad (9)$$

$$P_{max}^q(\diamond^{\leq i} G) = \max_{a \in A} \sum_{q' \in Q} T(q, a, q') P_{max}^{q'}(\diamond^{\leq i-1} G), \quad (10)$$

$$\forall q \in Q^?, i \in \{1, \dots, H\}.$$

Algorithm 2: Cost-bounded adaptive multi-stage sampling (CB-AMS)

input : A state $q = (b_s, e)$ in \mathcal{B} , the number of samples N_i , time horizon i .

output: The estimated maximal probability $\tilde{P}_i^{N_i}(q)$.

```
1 if  $e > D$  or  $i > H$  then
  | return 0;
  end
2 if  $b_s \in G$  then
  | return 1;
  end
3 for  $a \in A$  do
4   Sample a next state  $q' = (b', e')$  by taking action  $a$ ,
      where  $e' = e + C(s, a);$ 
5    $N_{a,i}^q = 1;$ 
6    $\tilde{Q}(q, a) = \text{CB-AMS}(q', N_{i+1}, i + 1);$ 
  end
7  $n = |A|;$ 
8 while  $n < N_i$  do
9    $a^* = \text{argmax}_a (\frac{\tilde{Q}(q,a)}{N_{a,i}^q} + \sqrt{\frac{2 \ln n}{N_{a,i}^q}});$ 
10  Sample a next state  $q' = (b', e')$  by taking action
      $a^*$ , where  $e' = e + C(s, a^*);$ 
11   $\tilde{Q}(q, a^*) = \tilde{Q}(q, a^*) + \text{CB-AMS}(q', N_{i+1}, i + 1);$ 
12   $N_{a^*,i}^q = N_{a^*,i}^q + 1, n = n + 1;$ 
  end
13  $\tilde{P}_i^{N_i}(q) = \frac{1}{N_i} \sum_a \tilde{Q}(q, a);$ 
14 return  $\tilde{P}_i^{N_i}(q);$ 
```

Then it can be seen that for $i \in \{1, \dots, H\}$,

$$\mu_i^*(q) = \text{argmax}_{a \in A} \sum_{q' \in Q} T(q, a, q') P_{max}^{q'}(\diamond^{\leq i-1} G).$$

B. Approximate Solution via Adaptive Sampling in Belief Space

As can be seen in Algorithm 1, the exact solution involves constructing the belief MDP \mathcal{B} , whose state space grows exponentially with the time bound H . It leads to the well-known curse of dimensionality in the computation of $P_{max}(\diamond^{\leq H} G)$ and hinders the application of Algorithm 1 to problems with large state spaces and a large time horizon H . Furthermore, the memory needed to store \mathcal{B} grows exponentially as well.

To address both the complex belief MDP \mathcal{B} and intractable computation of $P_{max}(\diamond^{\leq H} G)$, we now propose to use sampling algorithms [15] to estimate the optimal classification probabilities. In particular, we leverage the adaptive multi-stage sampling algorithm (AMS) proposed in [16]. The key observations for the active classification problem in (5) that make the AMS a reasonable choice are as follows. First, since the MDP models in \mathcal{M} are typically smaller than the belief MDP \mathcal{B} , it is easier to simulate sample paths in \mathcal{B} than explicitly specifying \mathcal{B} itself. Furthermore, AMS is particularly suitable for models with a large state space but small action space [16], where it is unlikely to revisit the

same belief state multiple times in a sampled run, which is exactly the case the belief MDP \mathcal{B} obtained with Algorithm 1. It can be observed that in \mathcal{B} , the action space remains the same as the original MDPs. Furthermore, the belief states in \mathcal{B} takes values in a continuous space where generally it is very rare to revisit the same belief state with the same accumulated cost in a simulated run. Algorithm 2 shows the belief state sampling procedure, termed as CB-AMS short for cost-bounded adaptive multi-stage sampling.

The input to Algorithm 2 is a state $q = (b, e)$ in \mathcal{B} , the number of N_i samples to be collected and the current time step i . The output is $\tilde{P}_i^{N_i}(q)$ which is the estimated maximal probability to reach a classification decision from state q with horizon $H - i$. The initial call is CB-AMS($(b_0, 0), N_0, 0$) is for the initial belief b_0 and time horizon 0. At Line 1, if the accumulated cost or the time horizon exceeds the bound, it will return 0, since the probability to reach a classification decision is 0, and there is no need to go further. At Line 2, if the belief b reaches its goal G , the algorithm will return 1. Then from Line 3 to Line 6, an initialization is performed to first try each action a and sample a subsequent state $q' = (b', e')$, where b' is sampled based on the transition probability as defined in (1). At Line 6, CB-AMS is called recursively where $\tilde{Q}(q, a)$ denotes the sum of returned rewards (essentially the probabilities) by executing action a from state q . In Line 7, n denotes the number of samples collected and is initialized to be $|A|$. We will then enter the sampling loop that terminates when the number of samples n reaches N_i . In each sampling iteration, we first select the action by the equation defined in Line 9. The selection criterion balances between sampling actions with a high average return value $\frac{\tilde{Q}(q, a)}{N_{a,i}^q}$ and trying actions that are less sampled as denoted by $\sqrt{\frac{2 \ln n}{N_{a,i}^q}}$, where $N_{a,i}^q$ denotes the number of times that the action a has been sampled from the state q at time horizon i . Obviously $\sum_a N_{a,i}^q = n$. It can be observed that the actions that lead to a higher probability to reach a classification decision are sampled more often and thus avoid exploring the belief states and actions that are unlikely to lead to a classification decision.

The following theorem shows that the output of Algorithm 2 converges to $P_{max}(\diamond^{\leq H} G)$ as the number of samples $N_i, 0 \leq i \leq H$ goes to infinity.

Theorem 1. *If CB-AMS gets run with input N_i for $i = 0, \dots, H$ with arbitrary initial condition $q \in Q$, then*

$$\lim_{N_0 \rightarrow \infty} \lim_{N_1 \rightarrow \infty} \dots \lim_{N_H \rightarrow \infty} E[\tilde{P}_0^{N_0}(q)] = P_{max}^q(\diamond^{\leq H} G),$$

where $P_{max}^q(\diamond^{\leq H} G)$ represents the maximum probability to reach the decision region G in H steps with costs no larger than D from state q .

Proof. In Algorithm 2, we are effectively sampling from the MDP \mathcal{B} with the state $q = (b_s, c)$ at time horizon i . From Line 1 and 2, once c exceeds the cost bound D or i exceeds the horizon bound H , no reward will be returned.

Otherwise, once b_s reaches the goal G within the cost and horizon bound, a reward of 1 will be returned. We denote $R(q, a)$ as the reward by executing action a at state q . Note that this reward is not to be confused with the cost function C that represents the classification cost, for example, the test and treatment costs for medical diagnosis. Equivalently, we assign $R(q, a) = 1$ for all $a \in A, q = (b_s, c), c \leq D, b_s \in G$ and $R(q, a) = 0$ for q 's otherwise. Given a strategy $\mu = \{\mu_t | \mu_t : Q \rightarrow A, 0 \leq t \leq H\}$, the value function $V_i^\mu(q)$ for state q and time step i is

$$V_i^\mu(q) = E\left[\sum_{t=i}^H R(q, \mu_t(q_t)) | q_i = q\right]$$

with $q \in Q, i = 0, \dots, H$. Given a state $q = (b_s, c), V_i^\mu(q) = 0$ if $c > D$ and $V_i^\mu(q) = 1$, if $c \leq D$ and $b_s \in G$. $V_{H+1}^\mu(q) = 0$. Furthermore, once reaching a state $q = (b_s, c)$ with $c > D$ or $b_s \in G$, the algorithm will return the corresponding reward and such q will not have successive states.

$V_i^\mu(q)$ can be equivalently written as

$$V_i^\mu(q) = \sum_{q' \in Q} P(q, \mu_i(q), q) V_{i+1}^\mu(q').$$

Therefore, it is not hard to see that $V_i^\mu(q) = P_q^\mu(\diamond^{\leq H-i} G)$. Algorithm 2 serves to approximate the optimal value function $V_i^* = \max_\mu V_i^\mu$ (equivalently the optimal probability).

Once we get the MDP with the reward structure R , where the value function $V_i(q)$ essentially denotes the probability to reach G from q within $H - i$ steps and the bounded cost, the rest of proof follows Theorem 3.1 in [16]. \square

Once the optimal value function (probability) has been estimated by Algorithm 2, it is then possible to extract the policy at each state q and horizon i by

$$\mu_i(q)^* = \operatorname{argmax}_a \sum_{q' \in Q} P(q, a, q) \tilde{V}_i^*(q'),$$

where $\tilde{V}_i^*(q) = \tilde{P}_i^{N_i}(q)$.

Note that this sampling approach is based on the given POMDP model, which has been obtained from history data, for example, the database of medical diagnosis. Therefore, at Line (4) and Line (10) of Algorithm 2, we sample from known distributions as defined by the POMDP model \mathcal{P} , instead of trying medication actions to patients and observe their reactions.

C. Reach-Avoid in Belief Space

Algorithm 1 and Algorithm 2 can be generalized naturally for (6) as the optimization objective, where the belief b should remain in $B_{safe} \subseteq B$. The unfolding procedure is almost identical with Algorithm 1 except in Line 14. There will be an additional constraint, where $b' \in B_{safe}$, i.e., if b' is not in B_{safe} , it will not be expanded in the next iteration. For Algorithm 2, the difference is in Line 1, where if $b \notin B_{safe}$, it will return 0.

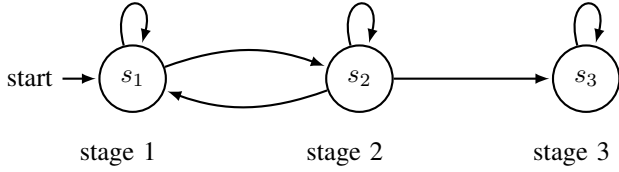


Fig. 2: Feasible state transitions in HMMDP.

V. SIMULATION

A. Medical Diagnosis and Treatment

In this section, we introduce a particular example in medical diagnosis, where HMMDP model is used to capture how stages for a family of diseases evolve based on tests and treatment as shown in Figure 2. In particular, the states in the MDP represents the early stage, medium stage and late stage of the disease, with an increasing order of the severity. There are two possible diseases modeled by two MDPs \mathcal{M}_1 and \mathcal{M}_2 that need to be diagnosed and treated. The action space includes three actions, namely

- 1) a_1 for treatment 1,
- 2) a_2 for treatment 2,
- 3) a_3 for doing nothing but observe,

where treatment i as denoted by a_i , is more effective on the disease i with costs $C(a_i)$. For a_3 , it does not incur any cost but the disease may have a higher probability to evolve to a later stage. The transition probabilities are as shown in the following matrices (11), where $T_i(a)(j, k) = T_i(s_j, a, s_k)$. The costs are as defined in (12) where $C(i, j) = C(s_i, a_j)$.

$$\begin{aligned}
 T_1(a_1) &= \begin{bmatrix} 0.8 & 0.2 & 0 \\ 0.7 & 0.2 & 0.1 \\ 0 & 0 & 1 \end{bmatrix}, T_1(a_2) = \begin{bmatrix} 0.6 & 0.4 & 0 \\ 0.2 & 0.4 & 0.4 \\ 0 & 0 & 1 \end{bmatrix}, \\
 T_1(a_3) &= \begin{bmatrix} 0.5 & 0.5 & 0 \\ 0.1 & 0.6 & 0.3 \\ 0 & 0 & 1 \end{bmatrix}, T_2(a_1) = \begin{bmatrix} 0.6 & 0.4 & 0 \\ 0.1 & 0.5 & 0.4 \\ 0 & 0 & 1 \end{bmatrix}, \\
 T_2(a_2) &= \begin{bmatrix} 0.9 & 0.1 & 0 \\ 0.8 & 0.1 & 0.1 \\ 0 & 0 & 1 \end{bmatrix}, T_2(a_3) = \begin{bmatrix} 0.3 & 0.7 & 0 \\ 0.1 & 0.3 & 0.6 \\ 0 & 0 & 1 \end{bmatrix},
 \end{aligned} \tag{11}$$

$$C = \begin{bmatrix} 2 & 5 & 0 \\ 6 & 4 & 0 \\ 7 & 7 & 0 \end{bmatrix} \tag{12}$$

The diagnosis decision is made for disease 1 or 2 if one of following is satisfied.

$$b_s(1) \geq \lambda_1 \text{ or } b_s(2) \geq \lambda_2, \tag{13}$$

with the initial belief $\hat{b}_{s_1} = (0.5, 0.5)$, cost constraint $D = 10$. One step unfolding according to the Algorithm 1 is shown in Figure 3. If $\lambda_1 = 0.8, \lambda_2 = 0.7$, it can be seen that if a_2 is executed, there is 0.25 probability that the disease is diagnosed to be type 1 (since $b_{s_2}(1) = 0.8$) at the shaded state q_4 , with a cost of 5. Therefore, q_4 will not be included in the states to be expanded in the next iteration.

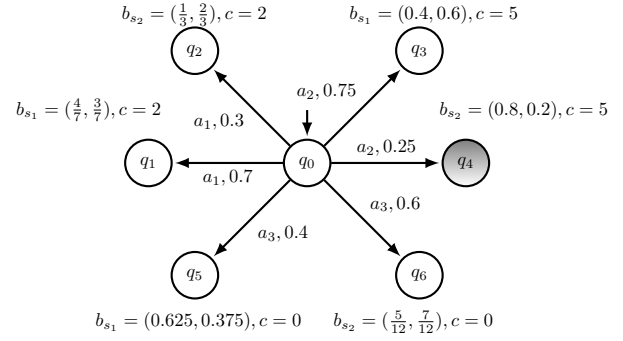


Fig. 3: One step unfolding

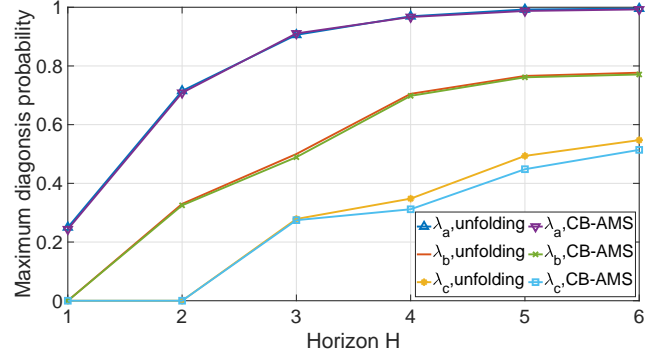


Fig. 4: Maximum probability for (7). $\lambda_a : \{\lambda_1 = 0.8, \lambda_2 = 0.7\}$, $\lambda_b : \{\lambda_1 = 0.9, \lambda_2 = 0.8\}$, $\lambda_c : \{\lambda_1 = 0.95, \lambda_2 = 0.9\}$.

We implement both Algorithm 1 and Algorithm 2 in C++. For Algorithm 1, the resulting MDP model is input into the PRISM model checker [17] to compute the maximum probability (7). For Algorithm 2, we set $N_i = 2000, 0 \leq i \leq H$. We also store the calculated values of $\tilde{P}_i^{N_i}(q)$ to avoid recomputing them. The results are as shown in Figure 4 to illustrate how the maximum probability to diagnose the disease increases with the horizon H . Typically, the higher the classification threshold, the less likely that belief states to make a classification decision can be encountered and thus there is less probability of successful diagnosis. Furthermore, the CB-AMS algorithm is able to closely estimate the optimal probability.

Now let's look at (6) where we try to reach a classification decision without visiting undesired belief states. For the medical diagnosis example, the desired property is to diagnose the disease without reaching the late stage of the disease, where the corresponding B_{safe} can be defined as

$$B_{safe} = \{b_s | s \neq s_3\}. \tag{14}$$

For the medical diagnosis example, the results are as shown in Figure 5. The same trend as in Figure 4 can be observed where the maximum probability to safely reach G without going out of B_{safe} increases with a longer time horizon. However, due to the extra safety constraint, the maximum probabilities also decreases, compared to the results without safety constraint. Again, the CB-AMS algorithm performs well to estimate the optimal probability.

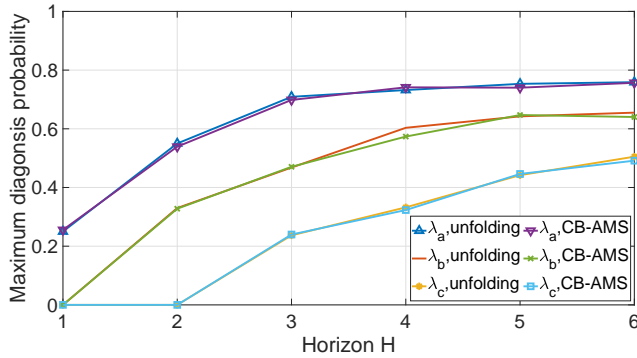


Fig. 5: Maximum probability for (6) with B_{safe} defined in (14). $\lambda_a : \{\lambda_1 = 0.8, \lambda_2 = 0.7\}$, $\lambda_b : \{\lambda_1 = 0.9, \lambda_2 = 0.8\}$, $\lambda_c : \{\lambda_1 = 0.95, \lambda_2 = 0.9\}$.

TABLE I: Run times for (7) in seconds

Horizon H		1	2	3	4	5	6
λ_a	Unfold	0.31	0.93	3.13	9.88	24.24	224.12
	CB-AMS	0.38	1.54	4.84	17.01	50.74	133.90
λ_b	Unfold	0.01	1.00	4.74	21.14	131.06	930.42
	CB-AMS	0.49	3.83	5.62	25.52	78.25	196.18
λ_c	Unfold	0.01	0.06	4.48	20.24	189.67	1418.75
	CB-AMS	0.40	1.71	5.54	26.40	103.84	272.88

The run times for both algorithms with regards to specification (7) and (6) are as shown in Table I and Table II. For Algorithm 1, the run time consists of the time to get the belief MDP model \mathcal{B} and compute the optimal probability. All the experiments were run on a laptop with 2.6GHz i7 Intel® processor with 16GB memory. It can be seen that for a small time horizon H , exact solution outperforms sampling in time consumption, as the number of the states in \mathcal{B} is small. But as the horizon H grows, the run time from sampling grows slower than the exact horizon and thus is more favorable.

TABLE II: Run times for (6) in seconds

Horizon H		1	2	3	4	5	6
λ_a	Unfold	0.40	0.71	2.39	7.62	30.34	229.89
	CB-AMS	0.37	1.38	4.26	14.39	39.48	98.97
λ_b	Unfold	0.01	0.92	3.04	12.31	94.13	1054.71
	CB-AMS	0.37	1.40	6.11	21.67	73.12	235.12
λ_c	Unfold	0.01	0.12	4.35	17.76	180.36	2211.70
	CB-AMS	0.39	1.72	6.20	28.87	91.46	330.35

B. Intruder Classification

In automated surveillance applications [18], it is often necessary to determine whether a detected target is a potential threat before deploying security resources for further intervention.

We present an example of a 8×8 gridworld shown in Figure 6. The target is not allowed to reach the green zone. The target is assumed to be either a hostile human intruder (Class 1) or an animal (Class 2) which has no real threat. The behaviors of the hostile and safe intruders are characterized by two MDPs \mathcal{M}_1 and \mathcal{M}_2 , respectively. The state in the MDPs refers to target's location in the gridworld, which is observable through radar or some other static sensors in the environment. At each time step the target moves to one of

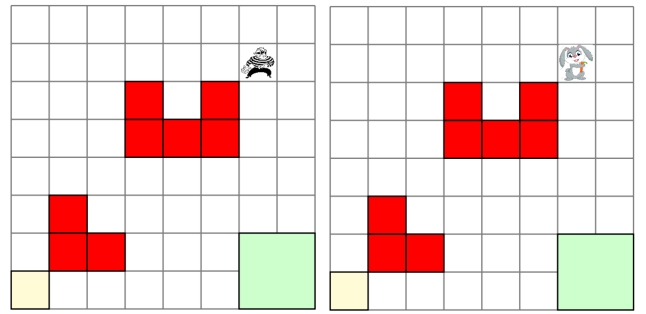


Fig. 6: Gridworld with two potential classes of targets - hostile (left) and safe (right). Green cells are sensitive areas. Red cells are obstacles such as buildings. The yellow cell is a hiding place for the hostile intruder.

its neighbouring cells randomly. The actions available to the automated surveillance are

- 1) a_1 for passive observation,
- 2) a_2 for alarm through loudspeakers.

If a_1 is chosen, a human intruder will attempt to move to the sensitive area (the green region) as denoted by S_{green} . When a_2 is executed, the animal will be startled and move in all directions with equal probability, while the human intruder will tend to move towards the yellow region in Figure 6 ostensibly to hide. The human moves randomly but generally heads to S_{green} or yellow region for action a_1 or a_2 . The randomness is to capture different human preferences and human's inherent decision uncertainty. The costs for a_1 and a_2 at each state are 1 and 3, respectively.

The corresponding B_{safe} can be defined as

$$B_{safe} = \{b_s | s \notin S_{green}\}.$$

The classification decision is made if one of following is satisfied.

$$b_s(1) \geq 0.7 \text{ or } b_s(2) \geq 0.7,$$

with the initial belief $\hat{b}_s = (0.5, 0.5)$, step bound $H = 6$ and cost bound $D = 8$, where \hat{s} is the initial state that intruders at as seen in Figure 6.

The results of the experiment can be seen in Figure 7, where exact solution is used to compute the optimal classification strategy. Suppose the target is a hostile intruder, Figures 7a- 7d illustrate a run of the human movement, where action a_1 is executed at $t = 0, 3, 6$ and action a_2 is executed at $t = 5$. Figure 7i-7l depicts to the corresponding beliefs.

In the scenario shown in Figure 7c, the target is near S_{safe} and the corresponding belief is as seen in Figure 7k which favors class 1. The optimal action at the time instance $t = 5$ is to sound the alarm. Then at $t = 6$, it is observed that the target moves towards the yellow cell and the belief of class 1 exceeds the threshold. This is where the classification terminates and a human operator will be alerted. Figures 7e-7h and 7m-7p shows the classification with a safe target (an animal) where similar behavior can be observed. After the alarm is used in $t = 5$ in Figure 7o, the rabbit runs to the top right corner as seen in Figure 7p, a very unlikely move for a

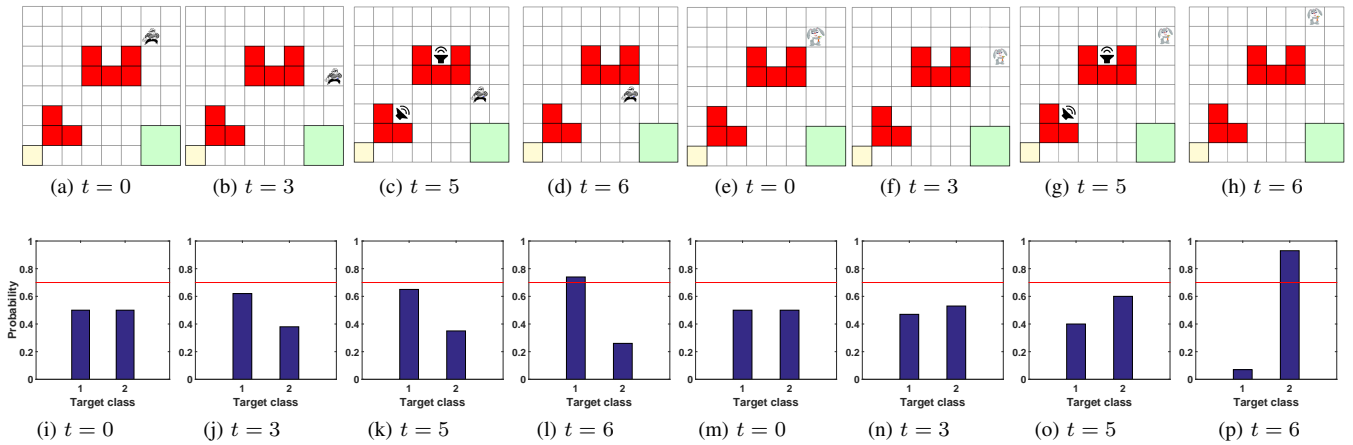


Fig. 7: Simulation of the classification problem in a gridworld environment with both target classes. Figures 7a- 7d and 7i-7l correspond to a simulation with a hostile target. Figures 7e- 7h and 7m-7p correspond to a simulation with a safe target.

hostile intruder. As a result, the belief for class 2 exceeds the threshold and the classification process terminates. In both simulations, the final costs are 7.

VI. CONCLUSION

In this paper, we studied a cost-bounded active classification of dynamical systems belonging to a finite set of MDPs. We utilized the POMDP modeling framework and the objective was to actively select actions based on the current belief, accumulated cost, and time step, such that the probability to reach a classification decision within a cost bound can be maximized. To solve the problem, we proposed two approaches. The first one was an exact solver to obtain the unfolded belief MDP model considering the cost-bound, and then compute the optimal strategy. To mitigate the computation burden, the second approach adaptively samples the actions to estimate the maximum probability. One bottleneck of the propose approaches is the computational complexity. In the future, we will explore point-based methods [19] that are popular in POMDP literature to surmount this difficulty.

REFERENCES

- [1] G. A. Hollinger, U. Mitra, and G. S. Sukhatme, "Active classification: Theory and application to underwater inspection," in *Robotics Research*. Springer, 2017, pp. 95–110.
- [2] M. Hauskrecht and H. Fraser, "Planning treatment of ischemic heart disease with partially observable markov decision processes," *Artificial Intelligence in Medicine*, vol. 18, no. 3, pp. 221–244, 2000.
- [3] T. Ayer, O. Alagoz, and N. K. Stout, "Or foruma POMDP approach to personalize mammography screening decisions," *Operations Research*, vol. 60, no. 5, pp. 1019–1034, 2012.
- [4] U. Adhikari, T. H. Morris, and S. Pan, "Applying hoeffding adaptive trees for real-time cyber-power event and intrusion classification," *IEEE Transactions on Smart Grid*, vol. 9, no. 5, pp. 4049–4060, 2018.
- [5] D.-S. Zois, M. Levorato, and U. Mitra, "Energy-efficient, heterogeneous sensor selection for physical activity detection in wireless body area networks," *IEEE Transactions on signal processing*, vol. 61, no. 7, pp. 1581–1594, 2013.
- [6] V. Myers and D. P. Williams, "A POMDP for multi-view target classification with an autonomous underwater vehicle," in *OCEANS 2010*. IEEE, 2010, pp. 1–5.
- [7] S. Ji, R. Parr, and L. Carin, "Non-myopic multi-aspect sensing with partially observable markov decision processes," *Target*, vol. 1, no. s3, p. s4, 2007.
- [8] M. T. Spaan, "Cooperative active perception using POMDPs," in *AAAI 2008 workshop on advancements in POMDP solvers*, 2008.
- [9] Q. Zhao, L. Tong, A. Swami, and Y. Chen, "Decentralized cognitive mac for opportunistic spectrum access in ad hoc networks: A POMDP framework," *IEEE Journal on selected areas in communications*, vol. 25, no. 3, 2007.
- [10] D. Fox, W. Burgard, and S. Thrun, "Active markov localization for mobile robots," *Robotics and Autonomous Systems*, vol. 25, no. 3-4, pp. 195–207, 1998.
- [11] Y. Wang, S. Chaudhuri, and L. E. Kavraki, "Bounded policy synthesis for POMDPs with safe-reachability objectives," in *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, 2018, pp. 238–246.
- [12] S. Andova, H. Hermanns, and J.-P. Katoen, "Discrete-time rewards model-checked," in *International Conference on Formal Modeling and Analysis of Timed Systems*. Springer, 2003, pp. 88–104.
- [13] M. L. Puterman, *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.
- [14] I. Chadès, J. Carwardine, T. G. Martin, S. Nicol, R. Sabbadin, and O. Buffet, "Momdps: a solution for modelling adaptive management problems," in *2012: Twenty-Sixth AAAI Conference (AAAI-12), Toronto, CAN, 2012-07-22-2012-07-26.*, 2012.
- [15] H. S. Chang, J. Hu, M. C. Fu, and S. I. Marcus, *Simulation-based algorithms for Markov decision processes*. Springer Science & Business Media, 2013.
- [16] H. S. Chang, M. C. Fu, J. Hu, and S. I. Marcus, "An adaptive sampling algorithm for solving markov decision processes," *Operations Research*, vol. 53, no. 1, pp. 126–139, 2005.
- [17] M. Kwiatkowska, G. Norman, and D. Parker, "Prism 4.0: Verification of probabilistic real-time systems," in *International conference on computer aided verification*. Springer, 2011, pp. 585–591.
- [18] S. Bharadwaj, R. Dimitrova, and U. Topcu, "Synthesis of surveillance strategies via belief abstraction," *arXiv preprint arXiv:1709.05363*, 2017.
- [19] W. S. L. Hanna Kurniawati, David Hsu, "SARSOP: Efficient point-based POMDP planning by approximating optimally reachable belief spaces," in *Proceedings of Robotics: Science and Systems IV*, Zurich, Switzerland, June 2008.