Safe Q-Learning on Complete History Spaces

Stephan Timmer and Martin Riedmiller

Neuroinformatics Group, University of Osnabrueck, Germany

Abstract. In this article, we present an idea for solving deterministic partially observable markov decision processes (POMDPs) based on a history space containing sequences of past observations and actions. A novel and sound technique for learning a Q-function on history spaces is developed and discussed. We analyze certain conditions under which a history based approach is able to learn policies comparable to the optimal solution on belief states. The algorithm presented is model-free and can be combined with any method learning history spaces. We also present a procedure able to learn history spaces especially suited for our Q-learning algorithm.

1 Introduction

In a POMDP setting, the learning agent does not have full information about the current state of the system. The common approach for solving POMDPs is therefore to replace the original state space S by the belief space of probability distributions over S. The belief space is huge and computing the optimal value function over beliefs can be very costly even for small state spaces. A still bigger problem is solving POMDPs without a model. In general, estimating the model (stochastic transitions and stochastic observations) is hard, since the current state $s_t \in S$ is unknown at any time step.

In this article, we investigate the technique of including short-time memory into the representation of the POMDP. By using short-time memory, we can maintain an estimate of the current state $s_t \in S$ of the system, which is adaptable with respect to size and precision. A reasonable way of establishing such an estimate is to consider a sequence of past observations and actions. Such a sequence is called a history list. In contrast to the history list approach, the belief space formulation of the POMDP is a perfect solution to the problem of estimating the current state, since all information about past events is merged into a probability distribution over states. However, due to the enormous complexity of learning policies on belief states, we restrict ourselves to suboptimal, but good policies that can be found with an algorithm based on history lists.

The overall problem of solving POMDPs without a model using history lists can be divided into two subproblems:

1. Building a set of history lists constituting a history space
2. Learning a policy on history lists after substituting the state space by the history space

First, we will concentrate on the second problem mentioned above, assuming that a history space is already available. We want to analyze certain conditions under which a history space can be used to learn near optimal policies by a variant of Q-learning. We will then present a procedure for learning history spaces, which is especially suited for our Q-learning algorithm. Unfortunately, it does not suffice to compute a Q-function on history lists and then extract a greedy policy. Before the Q-function can be exploited it is necessary to establish a sequence of observations and actions providing reliable information about the current state. We will discuss how such an approach relates to the optimal solution on belief states and present empirical evidence for a benchmark with more than one hundred states.

Although we will consider only deterministic systems, we choose to stay within the POMDP framework. Our work aims at solving reinforcement learning problems. The task of learning policies for deterministic POMDPs can be naturally formulated as such a problem.

2 Basic Facts About History Lists

Throughout the paper, we will assume a deterministic POMDP $(T, S, U, O, r, f, \Omega)$ such that T is a discrete set of time steps, S is a discrete state space, U is a discrete action space, O is a discrete observation space, $r : S \times U \rightarrow \mathbb{R}$ is the reward function, $f : S \times U \rightarrow S$ is a deterministic transition function and $\Omega : S \times U \rightarrow O$ is a deterministic observation model. To illustrate the use of history lists, consider the maze given in Figure 1. The agent, which is denoted by an arrow in a circle, is expected to find a certain goal cell in the maze. To determine the current position of the agent, only *bump-sensors* are available. These sensors, denoted by black dots, can tell the agent if there are walls at the four surrounding cells. The observation space therefore consists of sixteen combinations of walls plus an additional goal observation. The action space contains four distinct actions for moving left, right, up or down. If the agent tries to break through a wall or tries to leave the maze, the position of the agent remains the same.

How can the agent make use of history lists in this situation? The basic idea is to determine the current position of the agent by comparing history lists at

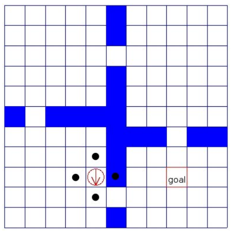

Fig. 1. Maze with partial observability

different time steps. If the same sequence of past wall observations and actions is given at two time steps $t \neq t'$, it is likely that also the position of the agent is the same at these time steps ($s_t = s_{t'}$). We do not expect a history list to contain the complete sequence of past observations and actions. In general, history lists can be of arbitrary length.

Definition 1. *History Lists*
A history list $h \in (U \times O)^*$ is a possibly empty sequence of action-observation pairs. The length $|h|$ of a history list is defined as the number of observations it contains. The space of possible history lists $H^* \subseteq (U \times O)^*$ contains all history lists which can be generated by the transition function f and the observation model Ω.[1]

Every time the agent notices a sequence of observations and actions that matches a history list $h \in H$ of a given history space $H \subseteq H^*$, an action u_h with respect to history h is executed. Assume that the sequence of observations and actions until time step $t \in T$ is given by $[u_0, o_1, u_1, o_2 ..., u_{t-1}, o_t]$. We call a history list $h \in H$ the current history list at time step $t \in T$ if h is a maximal suffix of this sequence with respect to the history space H. The current history list at time step t is denoted by h_t similar to the current state $s_t \in S$. To be compatible with the literature [1], we decided not to include reward signals into the definition of a history list. However, our algorithm will make use of reward signals.

A history list $h \in H$ is especially useful if it helps us to determine the current state $s_t \in S$ of the system. We will now give a more general definition of such sequences.

Definition 2. *Identifying History Lists*
A history list $h \in H^*$ is called identifying for a set of states $S' \subseteq S$, if $|h| > 0$ and the following proposition holds: If $[u_0, o_1, ..., u_{t-1}, o_t]$ is a sequence of observations and actions that can be generated by the transition function f and the observation model Ω, and h is a suffix of this sequence, then it holds that $s_t \in S'$.

In the following, we will use the term "identifying history list" only in the context of single states ($|S'| = 1$), unless the set S' is explicitly specified. What makes identifying history lists interesting is the fact that these lists are closed under some typical list operations.

Lemma 1. *Extension of History Lists (at the front)*
Let h be an identifying history list for a set of states $S' \subseteq S$. If h' is an extension of h such that h is a suffix of h', then h' is identifying for the set of states S'.

Proof. Let h' be a suffix of a possible sequence $[u_0, o_1, ..., u_{t-1}, o_t]$. Since h is a suffix of h', h is also a suffix of this sequence. Thus, it holds that $s_t \in S'$ because h is identifying for S'.

[1] If the observation model solely depends on the state ($\Omega : S \to O$) we allow history lists to have an additional observation at the front, representing the observation of the initial state of the history sequence.

Lemma 2. *Extension of History Lists (at the end)*
Let h be an identifying history list for a single state $s \in S$. Let h' be an extension of h such that an action $u \in U$ and an observation $o \in O$ is added at the end of h. If $h' \in H^*$, then h' is identifying for the single state $s' = f(s, u)$

3 Solving POMDPs with History Lists

Our algorithm consists of two separate modules.

1. At time step $t = 0$, a preferably short sequence of actions is executed, such that the current history list at time step $t' \geq 0$ becomes identifying for a single state. This corresponds to establishing a belief state in which a single state $s \in S$ has probability $p(s) = 1$, while all other states have probability zero. This module is called the efficient exploration strategy
2. From time step t' on, follow a greedy policy extracted from a Q-function defined on identifying history lists. This is reasonable, since every identifying history list corresponds to a single state $s \in S$. If at some time step later than t', the current history list becomes non-identifying, jump back to the first step of this enumeration (to reestablish an identifying history list)

To implement a procedure as described above, it is necessary to have a criterion able to decide whether a history list $h \in H$ is identifying or not. Furthermore, we need an efficient exploration strategy leading to an identifying history list as quickly as possible. We will now develop two criterions for detecting identifying history lists, as well as an efficient exploration strategy.

Definition 3. *Sufficient History Length*
Given a deterministic POMDP $M := (T, S, U, O, r, f, \Omega)$, the sufficient history length l_{su} is defined as
$l_{su} := max_{s \in S} \; min_{\, h \in H^*} \; \{|h|, h \text{ identifies state } s\}$

It is possible to find an identifying history list no longer than l_{su} for an arbitrary state $s \in S$. Thus, every history space $H \subseteq H^*$ should contain identifying history lists having a size at least equal to l_{su}. Otherwise, it would be impossible to identify every state in S.

Definition 4. *k-Complete History Space*
Let H_k denote the set of identifying history lists (for single states) which have a length less or equal to k. Let the set of minimal suffixes $H_k^{min} \subseteq H_k$ contain all identifying history lists $h \in H_k$ such that $\nexists h' \in H_k : h' \neq h$ and h' is a suffix of h. A history space $H \subseteq H^*$ is k-complete if $H_k^{min} \subseteq H$ and there are no identifying history lists $h, h' \in H, h \neq h'$ with $h \in H_{k-l_{su}}^{min}$ such that h is a suffix of h'.

A k-complete history space must contain an identifying history list h of length $|h| \leq k$, if h is a minimal suffix of an identifying history list. If h is such a suffix with length no longer than $k - l_{su}$, a k-complete history space must not contain extensions at the front of h.

Theorem 1. *Detection of Identifying History Lists*
Let $H \subseteq H^*$ be a history space and $h \in H$ a history list of length $\hat{k} := |h|$. If $k - \hat{k} \geq l_{su}$ for a fixed $k \in \mathbb{N}$, then the following two propositions hold:
H is k-complete : h is identifying $\Rightarrow \nexists h' \in H : h' \neq h$ and h is a suffix of h'
$H_k^{min} \subseteq H$: h is identifying $\Leftarrow \nexists h' \in H : h' \neq h$ and h is a suffix of h'

Proof. ($\Rightarrow$):
Let $h \in H$ be an identifying history list and let h^{min} be the minimal identifying suffix of h. Since $|h^{min}| \leq |h| = \hat{k} \leq k - l_{su} < k$ and H is k-complete, it must hold that $h^{min} \in H$. We assume that there exists a history list $h' \in H$ such that h is a suffix of h'. From Lemma 1, it follows that h' is an identifying history list. In this situation, H would not be k-complete since both h^{min} and h' are identifying history lists in H, $h^{min} \in H_{k-l_{su}}^{min}$ and h^{min} is a suffix of h'.
($\Leftarrow$):
Let $[u_0^*, o_1^*, ..., u_{t-1}^*, o_t^*]$ be a possible sequence of observations and actions and $[s_0, ..., s_{t-\hat{k}}, ..., s_t]$ be a corresponding sequence of system states.

Let $h := [u_0, o_1, ..., u_{\hat{k}-1}, o_{\hat{k}}]$ be a suffix of the above sequence. By definition of the sufficient history length l_{su}, there exists an identifying history list $h' \in H^*$, $h' := [u'_0, o'_1, ..., u'_{r-1}, o'_r]$ for state $s_{t-\hat{k}}$ with $r \leq l_{su}$. Consider the history list $h'' := [u'_0, o'_1, ..., u'_{r-1}, o'_r, u_0, o_1, ..., u_{\hat{k}-1}, o_{\hat{k}}]$ which is built by extending h' with history list h (at the end). It is easy to see that $h'' \in H^*$. By applying Lemma 2 several times, it follows that h'' is an identifying history list for state s_t. Furthermore, h'' has a length of at most k, since $|h| = \hat{k}$ and $|h'| \leq l_{su}$. Since $H_k^{min} \subseteq H$, there must exist an identifying history list $h''' \in H$ such that h''' is a minimal suffix of h'' and an identifying history list for state s_t . Since also h is a suffix of h'', it must hold that either h is a suffix of h''' or h''' is a suffix of h. Since we assume that there is no history list in H containing h as a suffix, h''' must be a suffix of h. Thus by Lemma 1, h is an identifying history list for s_t.

Note that it is not necessary to know the exact value of l_{su}. An overestimation of l_{su} will not affect the proof of the above theorem. It is easy to show that if the observation model only depends on the current state ($\Omega : S \rightarrow O$), the above theorem also holds for an extended history list $h = [o_0, u_0, .., u_{t-1}, o_t]$ including the observation of the initial state of the history sequence.

Since the criterion above can be checked solely by inspecting the history space H, it is possible to compute an efficient exploration strategy on the basis of H and a set of transitions $\mathcal{F}$ sampled from the underlying POMDP. These sample transitions can also be used to establish an alternative criterion: For a deterministic system, the sequence of observations and rewards is uniquely determined given a starting state and a sequence of actions. A history list h contains a link to all sampled episodes in which h became the current history list at some time t. By examining these episodes, it can be checked whether for an action sequence occurring after time t, the corresponding sequence of observations and rewards is uniquely determined. If this is the case for all action sequences (occurred after time t), the history list h is considered to be identifying. The idea for this

```
Safe Q-Learning on History Lists
Initialization
  1. $N := 0, \mathcal{F} = \emptyset, \forall h \in H, u \in U : \hat{Q}_0(h,u) := 0$
Main Loop
    1. Sample a set $\mathcal{F}_{new}$ of transition instances by applying an $\epsilon$-greedy exploration
        2. $\mathcal{F} = \mathcal{F} \cup \mathcal{F}_{new}$
        3. $\mathcal{F}^{vl} := \{(h_t, u_t, r_t, h_{t+1}) \in \mathcal{F} \mid u_t \in U^{vl}_{h_t}$ and $h_{t+1}$ is identifying $\}$
  Q-Learning Update Loop
    1. $\forall (h_t, u_t, r_t, h_{t+1}) \in \mathcal{F}^{vl}$
         $\hat{Q}_{N+1}(h_t, u_t) = r_t + \gamma \max_{u \in U^{vl}_{h_t}} \hat{Q}_N(h_{t+1}, u)$
        2. $N = N + 1$
  Until $\hat{Q}_N$ converges
End of Main Loop
```

Fig. 2. Q-Learning on Identifying History Lists

criterion was already introduced in [1] for building a prediction suffix tree, but without considering reward signals.

- To apply Q-learning on history spaces, two alternative criterions are available to detect identifying history lists.
  Criterion A A history list $h \in H$ is considered to be identifying if this can be proved by Theorem 1, assuming a k-complete history space or at least $H_k^{min} \subseteq H$. Additionally, h is considered to be identifying if it is an extension of another history list $h' \in H$ which can be proved to be identifying. Otherwise, h is considered to be non-identifying.
  Criterion B If the sequence of observations and rewards following a history list h is uniquely determined for action sequences occurred on sampled episodes, h is considered to be identifying. Otherwise, h has been proven to be non-identifying.
- Transition instances are represented by four tuples $(h_t, u_t, r_t, h_{t+1}) \in \mathcal{F}$, where r_t denotes the reward
- The set of valid actions $U_h^{vl} \subseteq U$ for history list h is empty if h is not identifying. Otherwise, the set U_h^{vl} contains action $u \in U$ if there exists a transition instance $(h_t = h, u_t = u, r_t, h_{t+1}) \in \mathcal{F}$ such that h_{t+1} is identifying
- The greedy action of $h \in H$ is computed with respect to the set of valid actions U_h^{vl}. If $U_h^{vl} = \emptyset$, a random action is taken

We implemented an efficient exploration strategy (Figure 3) trying to minimize the expected length of a path to an identifying history list. It would also be possible to develop a reward-based exploration which maximizes the rewards on a path to an identifying history list.

Theorem 2. *Safe Q-Learning on History Lists*
Let H be a history space. If criterion A is used to detect identifying history lists and it holds $H_k^{min} \subseteq H$, then the Q-function will converge to a unique fix point. If criterion B is used, then the Q-function will converge without any assumptions with respect to H.

Efficient Exploration Strategy
Input Set of sampled transitions $\mathcal{F}$, Q-function Q_N, current history list h_t
Algorithm

1. Estimate transition probabilities $p((h,u) \rightarrow h')$ for the history space H based on transition instances from $\mathcal{F}$.
2. If h_t is identifying, return the greedy action according to the function Q_N. If h_t is not identifying, compute an action sequence probably leading to an identifying history list. This can be implemented by considering action sequences of length $1 < l \leq l_{max}$ starting from the current history list $h_t \in H$. By inspecting the estimated transition model, every possible current history list $h_{t+l} \in H$ after l time steps can be discovered and added to a candidate set. Then, it can be checked by criterion A or criterion B, how many candidates are identifying history lists.

Fig. 3. Efficient Exploration Strategy

Proof. A detailed proof is omitted due to space constraints. If criterion A is used, convergence immediately follows from Theorem 1, since updates of the Q-function are made only on sequences consisting of identifying history lists. Since every identifying history list corresponds to a single state $s \in S$, the subset of the history space used to update the Q-function is markovian. If criterion B is used, then a similar argument holds: For an updated history list h, it holds that all stored sequences of observations and rewards (occurred on a sampled episode) are deterministic after history list h becomes the current history list. Thus, it is possible to create a new deterministic POMDP with a new state space S' and observation model Ω' such that all sampled episodes are compatible with the new POMDP and h is an identifying history list for a state in S'.

Now we want to discuss which criterion is better suited to detect identifying history lists. If a k-complete history space is available, it is preferable to choose criterion A, because it has been proven to work correctly and can be checked efficiently. Note that if the history space used is not k-complete, but at least it holds that $H_k^{min} \subseteq H$, then Q-learning will still converge, but the criterion will detect only a subset of all identifying history lists. If the history space used is of very poor quality, it seems to be better to choose criterion B. Criterion B gives only empirical evidence that a history list identifying, but at least guarantees convergence of the Q-learning algorithm. The problem with criterion B is that even if the Q-function converges, the extracted policy is not necessarily successful if applied to the problem. This comes from the fact that a history list classified as identifying by criterion B can actually be non-identifying. In such a situation, the efficient exploration strategy might lead to non-identifying history lists. Thus, we propose to consider a history list as identifying only if this is confirmed by both criterions. By this procedure we achieve guaranteed convergence and the performance of the exploration strategy will gradually improve with the quality of the history space used. The reason for this is that criterion B will never classify a history list as non-identifying if this is not truly the case but it will revise wrong classifications made by criterion A. In other words, if the criterions are combined, the resulting classification will always be better than a classification solely based on criterion A.

The question arises whether the learned Q-function gives a policy of high performance. It is easy to see that if H (or rather k) is sufficiently large, then the safe Q-learning algorithm will converge to a near optimal policy. This is due to the fact that every extension (at the end) of an identifying history list is again identifying (Lemma 2). If H contains identifying history lists corresponding to an optimal path from a starting state to the goal, Q-learning will investigate this path.

4 Empirical Results

We applied our algorithm to the partially observable maze shown in Figure 1. The size of the state space is $|S| = 104$, which is rather large compared to typical POMDP benchmarks from the literature. The reward is -1 at all non-goal cells. Note that for the maze considered it holds that $l_{su} = 5$. Since there are many cells in the maze having no walls surrounding them, it is necessary to take a couple of well advised steps to identify the current state. Since the observations only depend on the state, we used history lists with an additional (initial) observation at the front.

We compared our algorithm against the optimal solution on belief states computed by the Witness algorithm [2] and a random exploration strategy. The initial probability distribution over states (belief state) is chosen according to the initial observation of the sampled episode. By this procedure, we incorporate knowledge about the observation of the initial state of an episode into the belief state. Unfortunately, we were not able to compute the optimal policy on belief states for the whole maze. Since the state space contains more than one hundred states, the computational effort for establishing the optimal value function becomes intractable. We therefore conducted an additional experiment in which only the lower right part of the maze is considered. This reduced maze only has twenty states. For both mazes, we computed the optimal solution of the corresponding MDP (fully observable maze). This solution always bounds the optimal solution of the POMDP mazes.

The random exploration samples an action at random if it is not possible to prove that the current history list is identifying. Otherwise, it takes the greedy action according to the learned Q-function. The efficient exploration (Figure 3) tries to identify the current state as quickly as possible and then takes a greedy policy to the goal. Thus, the resulting policy is not optimal, but reasonably close to the optimum.

We conducted experiments with three k-complete history spaces such that each experiment corresponds to a different choice of the parameter $k \in \{6, 7, 8\}$. Only criterion A was used to detect identifying history lists, because for k-complete history spaces this criterion works perfectly (Theorem 1). All values presented are mean values from ten runs of the algorithm. Every run of an

Table 1. Performance of learned policies. The fourth/fifth column gives the number of steps to the goal averaged over every possible starting state. The values given in braces are the performance of the policy learned after the third iteration of the main loop.

Algorithm	History Space	Size ($\|H\|$)	Efficient	Random	Maze
MDP Opt.	-	-	2.2	-	small
POMDP Opt.	-	-	2.7	-	small
Safe Q	7-complete	3200	3.48	4.40	small
MDP Opt.	-	-	7.77	-	large
Safe Q	8-complete	119082	12.40 (13.83)	32.15 (34.75)	large
Safe Q	7-complete	30712	13.46 (19.37)	39.39 (40.36)	large
Safe Q	6-complete	14850	14.00 (22.78)	61.59 (55.17)	large

Table 2. Results of the second experiment. The setup of the experiment is equal to the first experiment (large maze, 8-complete) but with a reduced history space. The third (fifth) column of the last row shows an experiment in which all extensions of history lists of size six (seven) are deleted.

Deletion Method	# Deleted Lists	# Steps	# Deleted Lists	# Steps
Random	10000	12.52 (13.54)	20000	13.57 (14.59)
Random	30000	13.52 (22.76)	-	-
No Extensions	77432	13.08 (14.33)	96428	13.62 (14.19)

experiment (large maze) consists of fifteen iterations of the main loop of the safe Q-learning algorithm. In every iteration, 10^5 transition instances are collected by applying an ϵ-greedy policy. For the small maze, only a single iteration of the main loop is carried out, sampling a total number of 50000 transition instances. An episode sampled to collect transition instances ends if the goal state is reached or the number of collected transition instances exceeds the threshold max_step = 200. Sampling 10^5 transition instances approximately corresponds to 660 sampled episodes. The exploration rate is set to $\epsilon = 0.1$.

Table 1 shows that it is possible to learn good policies based on a Q-function defined on history lists. The performance of the algorithm develops with the size of the history space and with the length of the history lists, respectively. Moreover, the efficient exploration strategy achieves significantly better results than the random exploration strategy. The optimal solution on belief states for the large maze is presumably close to ten steps. To show the robustness of the algorithm against non k-complete spaces, we conducted an additional experiment with the efficient exploration strategy in which a number of identifying history lists are randomly selected from all lists having size eight (8-complete history space) and then deleted from the history space. To detect identifying history lists, we used a combination of criterions as discussed in the previous section. Table 2 shows that even if it holds that $H_k^{min} \not\subseteq H$, the algorithm still performs well.

5 Discussion

In this section, we want to discuss the applicability of history based approaches to real world problems. From our perspective, two questions are especially important in this context.

1. Is it possible to learn a k-complete history space?
2. Is it possible to scale the approach to stochastic POMDPs?

We performed preliminary experiments in which it was possible to learn k-complete history spaces by the following procedure based on sampling a number of episodes at random.

- Initially, the history space H consists of all possible history lists of length one (e.g. $h = uo$), $l := 0$
- Loop (until the history space H does no longer changes)
  - $l = l + 1$
  - Delete old data and collect new data by sampling a number of episodes. An episode is a sequence consisting of actions, observations and rewards.
  - Build all one step extensions (at the end and at the front) of all history lists in H having a size equal to l. One step means a single action-observation pair. If the size of an extended list does not exceed the threshold $k - l_{su}$, add the list to H.
  - If a history list in H is a (possibly multi step) front extension of another history list in H, which is classified as identifying by criterion B, then shorten the extended list by deleting the first (at the front) action-observation pair of the list
- End of loop
- Add all possible history lists to H having size $k - l_{su} < |h| \leq k$ except for front extensions of those identifying history lists, which were previously included during the loop (detected with criterion B)

The above procedure can be proven not to include front extensions of an identifying history list $h \in H$ with $|h| \leq k - l_{su}$. This is a critical aspect of learning a k-complete history space. We were able to learn a 8-complete history space for the large maze.

In the past, it has already been demonstrated that history list approaches are practical and can be scaled to complex problems, e.g. continuous, multidimensional state spaces ([3], [4]). In fact, even in cases in which it is not possible to learn k-complete history spaces, e.g. for infinite state spaces, the convergence proof for our algorithm still applies (criterion B). A certain benefit of k-complete history spaces is that the size of the history space learned can be adjusted by the parameter k. Even for small values of k, it will be possible to learn good policies on a k-complete space because of the efficient exploration strategy. While applying a policy, the efficient exploration strategy establishes exactly those identifying history lists for which stored Q-values are available.

We think that the basic idea of this approach can be carried over to the general case of stochastic POMDPs. It may always be advantageous to explicitly search for belief states in which the probability mass concentrates on few states. This is the reason why our definition of identifying history lists also covers sets of states. We think that a search procedure as mentioned above is much more efficient than computing Q-values for every single belief as it is done in classical POMDP algorithms. Following this idea, we currently work on an extension of our algorithm to sparse stochastic systems (systems with less stochasticity).

6 Related Work

This section contains a summary of existing work concerning the issue of learning POMDP representations with short-time memory as well as learning policies on those representations.

For a deterministic transition function and a deterministic observation model, a POMDP can be represented by a Finite State Automaton (FSA). To learn a model of an FSA, the concept of tests is introduced in [5]. A test itself consists of a sequence of actions, while the outcome of a test is defined as the observation made after executing the action sequence. A randomized algorithm is presented which is able to learn a sufficient number of tests such that future observations can be perfectly predicted. In [6], this idea is generalized to the stochastic case yielding Predictive State Representations (PSRs). In the context of PSRs, tests are considered to be sequences of observations and actions such that probabilities can be assigned to these sequences. Precisely, the expression $p(t|h) = p(o_1, .., o_k|hu_1, ..., u_k)$ for a test $t = u_1o_1...u_ko_k$ denotes the probability of observing a certain sequence of observations, given that a certain sequence of actions is executed and a certain history h (past events) occurred. Similar to the work in [5], it can be shown that the model of an arbitrary POMDP of finite size can be expressed by a sufficiently large set of tests $\{t_1, ..., t_n\}$ which is then called a PSR. The prediction vector $p(h) = [p(t_1|h), ..., p(t_n|h)]$ constitutes a sufficient statistic of the system. In [1], it is shown that for the deterministic case, it is possible to derive a PSR from a prediction suffix tree, which can be decomposed into a set of history lists. Existing literature about PSRs focuses only on predicting the state, but not on learning policies.

In [1], predictions of future observations of an FSA are made by building a prediction suffix tree with loops. A path through this tree starts at the root node and corresponds to a certain, possibly cyclic sequence of past observations and actions. The leaf nodes of the tree contain predictions about future observations. By allowing the tree to have loops, it is possible to deal with long subsequences of useless information which can be sandwiched into an enclosing sequence. This procedure can substantially reduce the number of history lists (branches of the tree) needed to be stored in memory. In our approach, we did not include loops, because loop detection is a potential source of error. An incorrectly detected loop (by inspecting sampled episodes) could damage the performance of the efficient exploration strategy. However, it would be easily possible to include the looping mechanism by inserting links from longer (identifying) history lists to shorter (identifying) history lists. In contrast to our algorithm, the suffix tree is only designed for prediction purposes. Neither an efficient exploration is used to identify the current state in minimal time nor is a policy learned. Moreover, the concept of k-complete history spaces makes it possible to bound the size of the history space by the parameter k. A suffix tree always provides perfect prediction, but is possibly of great depth.

In [3], a Q-learning variant is used to learn policies for POMDPs based on a tree structure called UTree. The tree is used to estimate the current state by distinguishing future rewards of sequences of observations and actions. Unfortunately, there is no guarantee that Q-learning will eventually converge in such a setting. Thus, UTree can only heuristically be applied to POMDPs.

7 Conclusion

We developed a variant of Q-learning able to learn near optimal policies for deterministic POMDPs without a model. This is accomplished by substituting the state space by a history space consisting of sequences of past observations and actions. The algorithm is sound in the sense that convergence can by achieved without any assumptions with respect to the history space used. By introducing the concept of k-complete history spaces, we showed that the performance of the algorithm gradually improves with the quality of the history space available. We empirically showed that the algorithm is robust against "incomplete" history spaces and a relatively simple algorithm is able to learn k-complete history spaces.

References

1. Holmes, M.P., Isbell, C.L.: Looping suffix tree-based inference of partially observable hidden state. In: Proceedings of the 23th International Conference on Machine Learning (ICML), Pittsburgh, Pennsylvania, USA, pp. 409–416 (2006)
2. Littman, M.L., Cassandra, A.R., Kaelbling, L.P.: Efficient dynamic-programming updates in partially observable markov decision processes. Technical Report CS-95-19, Brown University (1995)
3. McCallum, A.: Learning to use selective attention and short-term memory in sequential tasks. In: From Animals to Animats 4: Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior, pp. 315–324. The MIT Press, Cambridge (1996)
4. Timmer, S., Riedmiller, M.: Abstract state spaces with history. In: Proceedings of the 25th International Conference of NAFIPS, the North American Fuzzy Information Processing Society (2006)
5. Rivest, R.L., Schapire, R.E.: Diversity-based inference of finite automata. Journal of the Association for Computing Machinery 43, 555–589 (1994)
6. Littman, M., Sutton, R., Singh, S.: Predictive representations of state. In: Proceedings of the 14th International Conference on Neural Information Processing Systems (NIPS), pp. 1555–1561 (2002)