Decomposing Large-Scale POMDP Via Belief State Analysis

Xin Li, William K. Cheung, Jiming Liu
Department of Computer Science
Hong Kong Baptist University
{lixin,william,jiming}@comp.hkbu.edu.hk

Abstract

Partially observable Markov decision process (POMDP) is commonly used to model a stochastic environment with unobservable states for supporting optimal decision making. Computing the optimal policy for a large-scale POMDP is known to be intractable. Belief compression, being an approximate solution, reduces the belief state to be of low dimension and has recently been shown to be both efficient and effective in improving the problem tractability. In this paper, with the conjecture that temporally close belief states could be characterized by a low intrinsic dimension, a novel belief state clustering criterion function is proposed, which considers the belief states' spatial (in the belief space) and temporal similarities, resulting in belief state clusters as sub-POMDPs of much lower intrinsic dimension and to be distributed to a set of agents for collaborative problem solving. The proposed method has been tested using a synthesized navigation problem (Hallway2) and empirically shown to be able to result in policies of superior long-term rewards when compared with those based on only belief compression. Some future research directions for extending this belief state analysis appproach are also included.

1 Introduction

Markov decision process (MDP) is a well-known decision making framework under a stochastic environment. An MDP model consists of a finite set of states, a set of corresponding state transition probabilities and a reward function. Solving an MDP problem means finding an optimal policy which maps each state to an action so as to achieve the best long-term reward. One of the most important assumptions in MDP is that the state of the environment is fully observable. This, however, is unfit to a lot of real-world problems. Partially observable Markov decision process (POMDP) generalizes MDP in which the decision process is based on the incomplete information about the state. A POMDP model is essentially equivalent to that of MDP with the addition of a finite set of observations and a set of corresponding observation probabilities. Due to the partially observability on the environment, the observation history has to be memorized to help making proper decision, then the policy of a POMDP is a mapping from histories of observations to actions.

With compressing the observation history into a *belief* state, POMDP's policy is now defined over these belief state which is a probability distribution over the unobservable real states as an effective summary of the observation history. The policy of a POMDP is thus a mapping from a belief state to an action over the entire continuous belief space. The best bound of computing complexity for obtaining the exact solution of optimal policy is doubly exponential in the horizon [3]. For large-scale POMDP problems, it is computationally infeasible even though it is known that the value function can be proven piecewise linear and convex (PWLC) over the belief space [1].

In the literature, there exist a number of different methods proposed to solve large-scale POMDP problems efficiently, including the witness algorithm [1], VDC algorithm [10], BFSC alogrithm [11], etc. Another orthogonal direction for making the solution scalable is to take the divide-and-conquer approach, which, at the same time, can further facilitate the problem solving to be conducted in a multi-agent setting. While there have been some previous work on automatic decomposition of POMDP, efficient and effective paradigms to support POMDP decomposition and distribution are still lacking.

In this paper, we are inspired by the recently proposed belief compression approach and notice that analyzing a sample of belief states computed based on observations could in fact provide us a lot of hints for reducing the problem complexity in a problem-specific manner. For example, based on the fact that belief states of POMDP can typically be characterized by a much lower dimensional state space, the belief compression approach uses dimension reduction techniques like PCA and exponential PCA to reduce the problem complexity. With the conjecture that temporally

close belief states could be characterized by a set of clusters, each with a further reduced intrinsic dimension, this paper proposes to cluster belief states based on their spatial (in the belief space) and temporal similarities, resulting in belief state clusters as sub-POMDPs of much lower intrinsic dimension and to be distributed to a set of agents for collaborative problem solving. We have tested the proposed method using a synthesized navigation problem and showed that the belief state clustering approach can result in policies of superior long-term rewards when compared with those based on standard belief compression.

The remaining of this paper is organized as follows. Section 2 provides the background on belief compression. Section 3 describes the proposed belief state clustering technique. Section 4 provides the details for computing the sub-POMDP policies and how they are used for solving the entire POMDP as a whole. Experimental results are reported in Section 5 with possible extensions included in Section 6. Section 7 concludes the paper.

2 Belief Compression

Belief compression is a recently proposed paradigm [8], which reduces the sparse high-dimensional belief space to a low-dimensional one via projection. The principle behind is to explore the redundancy in computing the optimal policy for the entire belief space which is typically sparse. Using a sample of belief states computed based on observations of a specific problem, data analysis techniques like exponential principal component analysis (EPCA) can be adopted for characterizing the originally high-dimensional belief space using a compact set of belief state bases. This paradigm has been found to be effective in making POMDP problems much more tractable.

Let S denote the set of true states, $\mathbb{B}$ denote the belief space of dimension $|S|$, $b \in \mathbb{B}$ denote the belief state where its j^{th} element $b_i(j) \geq 0$ and $\sum_{j=0}^{|S|} b_i(j) = 1$, B denote a $|S| \times n$ matrix defined as $[b_1|b_2|...|b_n]$ where n is the number of belief states in the training sample.

According to [8], one can apply EPCA and obtain a $|S| \times l$ transformation matrix U which factors B into the matrices U and $\widetilde{B}$ such that

$$B \approx e^{U\tilde{B}} \tag{1}$$

where each column of B equals $b \approx b^r = e^{U\tilde{b}}$ and the dimension of $\widetilde{B}$ is $l \times n$. As the main objective of U is for dimension reduction, it is typical that $l << |S|$.

To compare with some standard dimension reduction techniques like PCA, EPCA is found to be more effective in dimension reduction. Also, EPCA can guarantee all the elements of a belief state to be positive, which is important as each belief state is a probability distribution by itself. However, the transformation is a non-linear one, making the value function of the projected belief states no longer piecewise linear. The consequence is that many existing algorithms taking the advantage of the piecewise-linear value function become not applicable together with belief compression. As suggested in [8], those sampled belief states in the projected space can be used as the states of a correspondingly formed MDP. One can then compute the optimal policy for that associated MDP.

3 Clustering Belief States for POMDP Decomposition

3.1 General Ideas

Rather than being yet another technique to address the POMDP's scalability issue, we perceive that the belief compression approach in fact opens up a new dimension for tackling POMDP problems. That is the possibility to apply data analysis techniques to the belief space, leading to the possibilities of having more elegant problem solving tricks.

3.2 Dimension Reduction Oriented Clustering

As mentioned in [8], the efficiency of belief compression is owing to its strategy for tackling the high-dimensional belief state which is one of the main causes for the exponential complexity. To further exploit the dimension reduction paradigm, we propose to decompose POMDP by analyzing the manifold of a set of sampled belief states for clustering. We anticipate that in these cases, there should exist some clusterings which could result in more substantial dimension reduction per cluster when compared with that of the overall belief states. In other words, the clustering criterion that we are looking for is one that is formulated to maximize the with-in cluster problem regularity to account for the further reduction. To contrast, most of the conventional data clustering techniques try to identify data clusters for maximizing the overall inter-cluster variance/distance while at the same time minimizing the overall intra-cluster variance/distance.

This idea can be intuitively interpreted as exploitation of the structural modularization from the belief state perspective. Thus, the proposed belief state clustering has some analogy with POMDP decomposition. However, in the literature, most of the proposed POMDP decomposition techniques focus on analyzing the original states of the POMDP, instead of based on the statistical properties of belief state occurrence as what being proposed in this paper.

3.3 A Spatio-Temporal Criterion Function for Clustering

In this paper, we propose to cluster the belief states based on both their euclidean distance as well as their temporal difference, with the conjecture that regularities should be easier to identify for temporally close belief states. Among all the clustering algorithms, the k-means algorithm [7] is here chosen just for the simplicity reason. It is based on a function defined for measuring the distance between the cluster means and each data item. Data found to be closest to one of the cluster means will contribute to the update of that mean in the next iteration. The whole process will repeat until it converges. For clustering belief states with the dimension-reduction objective, we define a spatio-temporal distance function between two belief states, given as

$$\begin{aligned} dist(b_i, b_j) &= \sqrt{dist_{spatial}(b_i, b_j) + \lambda dist_{temporal}(b_i, b_j)} \\ &= \sqrt{\|b_i - b_j\|^2 + \lambda \|\frac{i-j}{n|S|}\|^2} \end{aligned} \quad (2)$$

where λ is a trade-off parameter for controlling the relative contribution of the first (spatial) term and the second (temporal) term. $n|S|$ is introduced to normalize the second term to be within $[0, \frac{1}{n}]$. If λ is too large, it will dominate the first term and the k-means clustering results will essentially be cutting the belief state sample into some consecutive parts according to the belief state appearance sequence in the sample. Also, the value of k, i.e., the number of clusters, is another parameter that one can tune for optimal belief state dimension reduction. To determine the values of λ and k, we only used an empirical procedure in this paper to be explained in the subsequent section. It is in fact possible to replace the k-means clustering with methods like mixture of Gaussians so that the data partition becomes soft instead of hard and the analytical derivation of optimal λ could be possible. This part will further be pursued due to the promising empirical results we obtained in this paper.

As just mentioned, the optimality of k and λ should be defined based on some criterion function which measures the difference between the original belief states and the reconstructed belief states after belief compression is applied. As each belief state is a probability distribution, Kullback-Leibler (KL) divergence could be used for evaluating the discrepancy between the original belief states and the reconstructed belief states, as given in Eq.(3).

$$\overline{KL}(B) = \frac{\sum_{i=1}^{n} KL(b_i \| b_i^r)}{n} \quad (3)$$

$$KL(b_i \| b_i^r) = \sum_{j=1}^{|S|} b_i(j) \ln \left(\frac{b_i(j)}{b_i^r(j)} \right). \quad (4)$$

For the original belief compression, the compression is based on primarily one transformation matrix U as described in Section 2. Now, as the belief states are clustered, there will be several transformation matrices, each corresponding to a particular cluster. Let the belief state sample be partitioned into P clusters $\{C_1, C_2, ..., C_P\}$ and the transformation matrix of the p^{th} cluster C_p to be U_p. The reconstructed belief states associated to C_p can then be approximated as $b^{r,C_p} = e^{U_p \widetilde{b}}$. To measure the dimension reduction effectiveness via the clustering, the KL divergence per cluster is to be computed, given as

$$\overline{KL}(C_p) = \frac{\sum_{b_j \in C_p} KL(b_j \| b_j^{r,C_p})}{|C_p|}. \quad (5)$$

Before proceeding to the next section for computing the policy, we would like to highlight the fact that clustering the belief states can result not only in reducing the overall complexity for solving the original POMDP problem, but also that for performing the EPCA for belief compression and that for computing the transition probabilities of the projected belief states. This computational gain is achieved at the expense of the clustering overhead as well as the optimality of the resulting policy that we may sacrify after the problem decomposition. Fortunately, the clustering overhead is found to be not significant when compared with the overall complexity. For the resulting policy's optimalty, the results we obtained so far are very positive.

4 Computing POMDP Policy

As mentioned in Section 2, those existing efficient exact algorithms (e.g. Witness algorithm [1]) no longer fit to solve the POMDP problem with reduced dimension due to EPCA's non-linear projection. As in [8], we use the MDP value iteration method on the low-dimensional sampled belief states to get an approximate policy, which has been proven to be a bounded-error approximation in [4]. While we do not have major contribution in this part, related formulations are still repeated here for completeness.

Let $\widetilde{B}$ denote the set of belief state clusters, each being associated with a different transformation matrix U_p. Thus, we have

$$\widetilde{B} = \{\widetilde{B}^{C_1}, \widetilde{B}^{C_2}, ..., \widetilde{B}^{C_p}\} \quad (6)$$

where

$$\widetilde{B}^{C_i} = \{\widetilde{b}_j | b_j^r \in C_i\}. \quad (7)$$

The approximate value iteration algorithm uses the following rule to compute a t-step lookahead value function V^t from a $(t-1)$-step lookahead value function V^{t-1}, given as

$$V^t(\widetilde{b}_i^{C_k}) = max_a(\widetilde{R}^{C_k}(\widetilde{b}_i^{C_k}, a) + \gamma \sum_{\widetilde{b}_j^{C_k}} \widetilde{T}^{C_k}(\widetilde{b}_i^{C_k}, a, \widetilde{b}_j^{C_k}) \cdot V^{t-1}(\widetilde{b}_j^{C_k})) \quad (8)$$

where $\widetilde{R}^{C_k}$ and $\widetilde{T}^{C_k}$ are the approximate reward and transition functions in the corresponding partitioned low-dimensional space.

4.1 Computing the Reward Function

The reward function $R(s_i, a)$ denotes an immediate reward if taking an action a at state s_i. Naturally, an immediate reward after taking an action a at belief state b or $\widetilde{b}$ should be the expected value over the all true states. See following equation:

$$\widetilde{R}^{C_i}(\widetilde{b_j}, a) = \sum_{i=1}^{|S|} R(s_i, a) b_j(s_i) \quad (9)$$

Note that in some problems, there is another form of reward function $R(s_i, a, s_j)$ which means the immediate reward is also relative to the state to be reached. Also, we can get the expected $R(s_i, a)$ from $R(s_i, a, s_j)$,

$$\widetilde{R}(s_i, a) = \sum_{j=1}^{|S|} R(s_i, a, s_j) T(s_i, a, s_j) \quad (10)$$

4.2 Computing the Transition Function

Computing the transition function of the projected belief states is a bit more complicated. One should first recur to the transition trajectory of the high-dimensional space based on the Bayes rules. It is a process in and out of the high-dimensional and low-dimensional space to accomplish the beliefs' evolvement, projection, reconstruction and matching, as described in [8]. For our proposed method, we only consider pairs of low-dimensional beliefs in the same cluster, regardless of the possible transitions between clusters. Thus, we get the transition function $\widetilde{T}^{C_k}(\widetilde{b}_i^{C_k}, a, \widetilde{b}_j^{C_k})$ as the sum of $p(z, j|i, a)$ over all observations z, *i.e.*,

$$p(z, j|i, a) = \omega(\widetilde{b}_j^{C_k}, \widetilde{b}'^{C_k}) \sum_{l=1}^{|S|} p(z|s_l) b_a^{C_k}(s_l) \quad (11)$$

where $\widetilde{b}'^{C_k}$ is the low-dimensional belief projected from a high-dimensional belief b^{C_k} of a cluster C_k, which is updated after executing an action and receiving an observation from the high-dimensional reconstruction of $\widetilde{b}_i^{C_k}$ using $b^{r,C_k} = e^{U\widetilde{b}^{C_k}}$. $\omega(\widetilde{b}_j^{C_k}, \widetilde{b}'^{C_k}) = \frac{1}{k}$ as we use k-nearest-neighbor for approximate discretization on the low-dimensional belief space. Also, $p(z|s_l)$ in Eq.(11) can be computed as

$$p(z|s_l) = \sum_{i=1}^{|S|} p(a_i|s_l) p(z|a_i, s_l) \quad (12)$$

with $p(a_i|s_l) = \frac{1}{|Actions|}$ and $p(z|a_i, s_l)$ is the given observation probability. For $b_a^{C_k}(s_l)$ in Eq.(11), it denotes the expected belief and can be computed as

$$b_a^{C_k}(s_l) = \sum_{j=1}^{|S|} T(s_j, a, s_l) b_i^{C_k}(s_j). \quad (13)$$

It is updated only by executing an action instead of using both action and observation.

Generally speaking, constraining the transitions within clusters will bring some reward information loss and weaken the policy quality accordingly. However, the spatio-temporal clustering we adopted is essentially geared to reduce the loss to a certain extent since it is based on the conjecture that belief state visited within a short period will try to be clustered as far as possible based on the spatio-temporal notion. In other words, good clustering results should benefit not only dimension reduction, but also the accuracy of the subsequently computed policy.

4.3 Value Function Computation and Policy Application

The final step is to compute the value function for each cluster to get the policy tables corresponding to the clusters using the reward and transition functions computed according to the previous two subsections. Based on Eq.(8), the conventional MDP value iteration algorithm can be used, which will stop when the value at time step $t+1$ is mathematically close to the value at time step t.

To apply the policy in a multi-agent setting, the computed policy tables will be distributed to different agents and a coordinating agent is needed with the role of selecting which agents to forward a new observation to based on comparing the corresponding high-dimensional belief state with the sampled beliefs. Our implementation selects the nearest one which is indexed with the corresponding agent for taking the next step action based on the agent's policy table.

5 Experimental Results

5.1 The Hallway2 Problem

The Hallway2 Problem which is defined with a specific maze is commonly used to test the scalability of algorithms

for solving POMDP problems (see also [1]). The problem is to find the goals in the maze with 92 states (4 being the goal states), and contains 5 actions and 17 types of observations. Reaching one of the goal states will yield a $+1$ reward and then the next state will be set to a random non-goal state. In addition, it is assumed that all the non-goal states of the problem are equally likely to be the initial state location and thus the starting belief state is $b_1 = (\frac{1}{88}, ..., \frac{1}{88}, 0.0, 0.0, 0.0, 0.0, \frac{1}{88}, ..., \frac{1}{88})^T$. Also, the discount factor used is 0.95. In this paper, all the experimental results reported are based on this problem setting.

5.2 Belief State Sampling

The process of belief compression is operated on a belief state sample generated via simulation. During the simulation for sample generation, two levels of random numbers are used to select an action, and the Bayes rules are used to evolve the belief states. When one random number is found to be less than the threshold defined as 0.5, another random number will be generated to decide the next action. Otherwise, it will sum up all the beliefs generated so far and take the state with the maximal sum of probabilities as the current state. Then, an MDP solver will be called to get the corresponding policy table to choose the next action for its 'current state'.

Note that the sampled belief states in consecutive time steps often own the similar shape with the same number of modes [6]. Obviously, these "structural" similar belief states could have them represented at a much lower dimension. That's why the belief space is often considered to be sparse.

5.3 Performance of Belief State Clustering

The first experiment focuses on evaluating the effectiveness of the proposed spatio-temporal clustering scheme for overall dimension reduction. We enumerated a set of different values for the trade-off parameter λ as well as the number of the clusters P and evaluated the corresponding dimension reduction performance. For performance measurement, we contrasted the values of the KL-divergence between the set of original belief states and the ones reconstructed based on the conventional belief state compression (i.e., without clustering) and the one we proposed in this paper with belief state clustering. We only care those parameter settings (λ, l, P) on which the averaged KL divergence of each cluster is less than that using the conventional belief compression. It is noted that a number of settings can result in better overall dimension reduction. Among those settings, we set a filter and highlight those with high reduction. The filtering is based on a ratio R, defined as

	# Items	Original EPCA	Proposed Method	Comp. Cost (sec.)
$Cluster1$	96	1.3997	0.0024	1.84
$Cluster2$	16	0.4998	0.0004	0.34
$Cluster3$	36	0.1893	0.0003	0.49
$Cluster4$	352	4.2596	0.4938	69.27

Table 1. Performance comparison between the conventional EPCA for belief compression and the proposed method, where the number of clusters is 4, the reduced dimension is 3 and $\lambda = 3$.

$$R(\lambda, l, P) = 1/P * \sum_{p=1}^{P} \frac{(KL_U(\lambda, l, C_p) - KL_{U_p}(\lambda, l, C_p))}{KL_U(\lambda, l, C_p)} \quad (14)$$

where $KL_U(\lambda, l, C_p)$ stands for the KL-divergence between the original belief states in the p^{th} cluster and the corresponding reconstructed belief states based on only U (original EPCA), and $KL_{\{U_p\}}(\lambda, l, C_p)$ stands for the KL-divergence between the original belief states in the p^{th} cluster and the corresponding reconstructed belief states based on U_p (resulted from applying EPCA to the cluster).

We select a parameter setting (3,3,4) with $R > 0.95$. Table 1 tabulates the performance measures for comparing the KL divergence under this parameter setting. Figure 1 shows the comparison of the average KL-divergence over all sampled beliefs at different reduced dimensions using EPCA. Obviously, our proposed method achieves more accurate reconstruction than that of using conventional EPCA. In addition, as reported in the last column of Table 1, our proposed method took 71.94 seconds while the conventional EPCA took 153.08 seconds.

5.4 Policy Quality with Spatio-Temporal Clustering Introduced

In terms of those parameter settings (λ, l, P) with significant intrinsic KL-Divergence reduction, we compute the policy for each cluster and test the policy. The comparison of policy performance occurs between the computing policy via dimension reduction directly and computing policies via spatio-temporal clustering using different numbers of bases. For each parameter setting, we execute 1000 trials. Each trial is a trajectory with maximal 251 steps before any one of the objective states is reached. The trajectory is evolved by executing the computed policies. Our experimental results show that nearly half of these parameter settings result in the policy quality enhancement, and some of them help

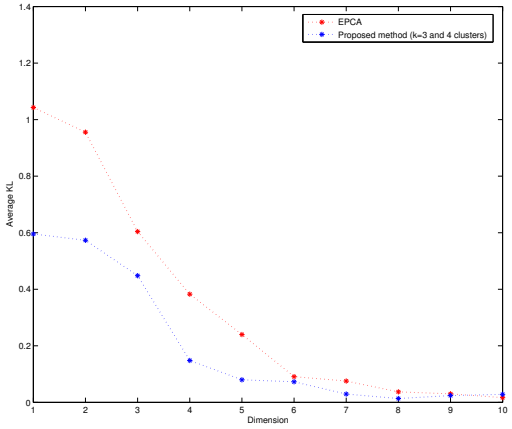

Figure 1. Average KL Divergence for conventional EPCA.

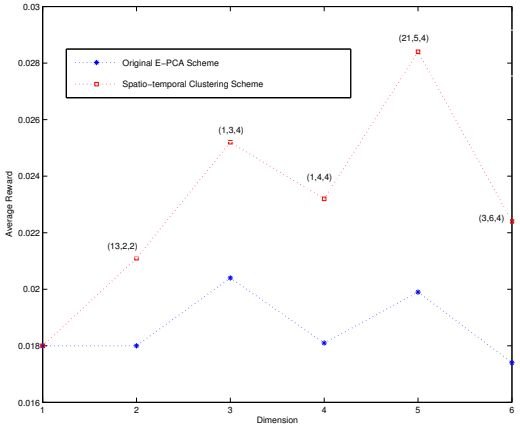

Figure 2. A comparison of policy performance using different schemes for average reward over 1000 trials with different parameter setting which is labelled as (λ, l, P).

increase the average reward greatly.

According to Figure 2, we can see obvious performance enhancement over the conventional belief compression. For the Hallway2 problem with 500 sampling beliefs, it also shows that using four clusters is a generally better strategy.

Table 2 lists out the detailed parameter settings selected for performance comparsion. Generally speaking, those settings with relatively higher $R(\lambda, l, P)$ ratio induces a better average reward, which is consistent to our conjecture that a clustering with a better dimension reduction power should also result in a policy of higher quality. It is also noted that it is hard to set a threshold of getting a good set of l and P as the value of the ratio $R(\lambda, l, P)$ for resulting in better policy varies quite a lot given different set of l and P.

As being discussed before, the performance enhancement is induced by the much more accurate low-dimensional representation, though some rewards among clusters are lost inevitablely. Our experimental results are consistent with what we have discussed in the previous section and show that the reward loss do not affect much the overall performance given good spatio-temporal clustering results.

6 Discussion and Future Works

This paper mainly demonstrates the possibility of clustering the belief states in a spatio-temporal manner for achieving further belief state compression and good policy performance. We are currently working on several extensions of this work as depicted as follows.

6.1 Towards Optimal Spatio-Temporal Clustering

While the criterion function used in this paper has shown to be effective empirically, it is by no means an optimal choice. In addition, we still lack automatic mechanisms (other than exhaustive search) for setting the parameters to govern the clustering. We believe that this is an immediate and important research direction to be pursued in the future.

6.2 The Multi-Agent Consideration

In this paper, we distribute each sub-POMDP to a problem solving agent. The agents are basically independent to each other, except to be coordinated by the brokering agent. As the decomposition based on the proposed belief state clustering may not result in a set of sub-POMDP problems which are equivalent to the original POMDP problems, interaction between those agents for achieving the overall optimal policy is an important research issue. Nash Equilibrium is an important concept commonly used in multi-agent learning [5] for solving decentralized MDP [2] and POMDP problems [9]. Our research agenda also includes how to apply this paradigm to the our decomposition scheme for further performance boosting. The basic idea is that every agent would conjecture other agents' behaviors and give the best response to other agents from its local view. A Nash equilibrium usually would not deduce the optimal policy. However, it should be able to guarantee a not-too-bad sub-optimal one.

avg. reward without clustering	avg. reward with clustering	λ	l	P	$R(\lambda, l, P)$
	0.0191	11	2	2	0.1781
0.0180	0.0211	13	2	2	0.3646
	0.0182	19	2	2	0.2167
	0.0180	3	2	4	0.7857
0.0180	0.0161	13	2	4	0.5229
	0.0149	21	2	4	0.4954
	0.0258	19	5	4	0.6387
0.0199	0.0284	21	5	4	0.7044
	0.0224	3	6	4	0.7857
0.0174	0.0214	17	6	4	0.5229
	0.0185	19	6	4	0.4954

Table 2. Performance comparision for different parameter settings.

What being described so far assumes that the whole model of the decision process is known. That is, we have the perfect knowledge about the reward function, transition function and observation function. Solving the corresponding POMDP problems is an off-line process. It is also interested to see how the multi-agent approach can be extend to support online learning (e.g., Q-learning [12]) for POMDP under partial observation scenarios.

7 Conclusion

This paper extends the recently proposed belief compression by introducing a spatio-temporal belief state clustering for addressing large-scale POMDP problems. It was found that the proposed spatio-temporal method can further compress the belief states in each cluster to a much lower dimension while maintaining similar belief state reconstruction accuracy and thus a better policy. Also, each cluster of belief states can naturely been distributed to different agents for collaborative problem solving. Future research directions include at least further enhancement in automatic determination of clustering parameters, hierarchical clustering of the belief states and the integration of the proposed belief state clustering and the multi-agent paradigm as a unified solution for solving large-scale POMDP problems.

Acknowledgements

We would like to thank the anonymous reviewers' valuable comments and insight into this work. This work has been partially supported by RGC Central Allocation Group Research Grant (HKBU 2/03/C).

References

[1] A.Cassandra. *Exact and approximate algorithms for partially observable Markov decision processes*. U.Brown, 1998.

[2] R. Becker, S. Zilberstein, V. Lesser, and C. V. Goldman. Transition-Independent Decentralized Markov Decision Processes. In *Proceedings of the Second International Joint Conference on Autonomous Agents and Multi Agent Systems*, pages 41–48, Melbourne, Australia, July 2003. ACM Press.

[3] D. Burago, M. de Rougemont, and A. Slissenko. On the complexity of partially observed Markov decision processes. *Theoretical Computer Science*, 157(2):161–183, 1996.

[4] G. J. Gordon. Stable function approximation in dynamic programming. In A. Prieditis and S. Russell, editors, *Proceedings of the Twelfth International Conference on Machine Learning*, pages 261–268, San Francisco, CA, 1995. Morgan Kaufmann.

[5] M. P. W. Junling Hu. Nash q-learning for general-sum stochastic games. *Journal of Machine Learning Research*, 4:1039–1069, 2003.

[6] X. Li, W. K. Cheung, and J. Liu. Towards solving large-scale pomdp problems via spatio-temporal brief state clustering. Edinburgh, Scotland, July 2005.

[7] J. MacQueen. Some methods for classification and analysis of multivariate observations. In *5th Berkley Symposium on Mathematics and Probability*, pages 281–297, 1967.

[8] N. Roy, G. Gordon and S. Thrun. Finding approximate POMDP solutions through belief compressions. *Journal of Artificial Intelligence Research*, 23:1–40, 2005.

[9] R. Nair, M. Tambe, M. Yokoo, D. Pynadath, and S. Marsella. Taming decentralized POMDPS: Towards efficient policy computation for multiagent settings. 2003.

[10] P. Poupart and C. Boutilier. Value-directed compression of POMDPS. In S. T. S. Becker and K. Obermayer, editors, *Advances in Neural Information Processing Systems 15*, pages 1547–1554. MIT Press, Cambridge, MA, 2003.

[11] P. Poupart and C. Boutilier. Bounded finite state controllers. In S. Thrun, L. Saul, and B. Schölkopf, editors, *Advances in Neural Information Processing Systems 16*. MIT Press, Cambridge, MA, 2004.

[12] C. Watkins. *Learning from Delayed Rewards*. PhD thesis, Cambridge Univ., Cambridge England, 1989.