A Novel Orthogonal NMF-Based Belief Compression for POMDPs

Xin Li LIXIN@COMP.HKBU.EDU.HK
William K. W. Cheung WILLIAM@COMP.HKBU.EDU.HK
Jiming Liu JIMING@COMP.HKBU.EDU.HK
Zhili Wu VINCENT@COMP.HKBU.EDU.HK
Department of Computer Science, Hong Kong Baptist University, Kowloon Tong, HK

Abstract

High dimensionality of POMDP's belief state space is one major cause that makes the underlying optimal policy computation intractable. Belief compression refers to the methodology that projects the belief state space to a low-dimensional one to alleviate the problem. In this paper, we propose a novel orthogonal non-negative matrix factorization (O-NMF) for the projection. The proposed O-NMF not only factors the belief state space by minimizing the reconstruction error, but also allows the compressed POMDP formulation to be efficiently computed (due to its orthogonality) in a value-directed manner so that the value function will take same values for corresponding belief states in the original and compressed state spaces. We have tested the proposed approach using a number of benchmark problems and the empirical results confirms its effectiveness in achieving substantial computational cost saving in policy computation.

1. Introduction

Partially Observable Markov Decision Process (POMDP) models how an agent acts in a stochastic environment given partial observations and feedback from the environment for a better average reward in the long run. Due to the partial observability, it is common to represent the *belief state* of a POMDP as a probability mass function defined over the true states. Upon each action-taking and then observation arrival, the belief state is re-estimated using Bayesian updating. The complete set of belief states spans a $|S| - 1$ dimensional continuous hypercube (also called belief space). Solving a POMDP is equivalent to computing its optimal policy (a mapping between belief states and actions) which is known to be computationally challenging. Even though there exist related computational shortcuts, the complexity bound for obtaining the optimal one with t steps ahead considered has been shown to be $O(\zeta_{t-1}^{|Z|})$ (Cassandra, 1998) where Z is the set of possible observations and ζ_i is the space complexity of the value function at the i^{th} iteration. So, the higher the dimension of the belief space, the larger will be the value of ζ_i and thus the overall complexity.

Belief compression refers to the methodology which projects the high-dimensional belief space to a low-dimensional one for approximation in order to cut down the policy computation cost. In the literature, two main approaches have been proposed. One explores the belief space's sparsity by analyzing belief samples (N. Roy and G. Gordon & Thrun, 2005), and another derives the POMDP formulation in the compressed space in a valued-directed manner so that the value function will take same values for corresponding belief states in the original and compressed spaces (Poupart & Boutilier, 2003).

In this paper, we propose to combine the strengths of the two approaches via a novel orthogonal non-negative matrix factorization (O-NMF). The proposed belief compression approach has a number of advantages, including: (1) O-NMF guarantees all the elements of the low-dimensional belief states to be non-negative, which is important as belief states are by themselves probability distributions; (2) O-NMF explores sparsity in belief space; (3) The value-directed property can be maintained as far as possible; (4) The overhead computation needed for getting the compressed POMDP formulation is carefully designed to avoid solving LP problems; and (5) The high-dimensional α vectors (characterizing the value func-

Appearing in *Proceedings of the 24th International Conference on Machine Learning*, Corvallis, OR, 2007.

tion) can be recovered from their low-dimensional counterparts in a well-posed manner. We have demonstrated the effectiveness of the proposed compression by applying it to a number of benchmark problems[1].

2. POMDPs and Belief Compression

2.1. POMDP Basics

A POMDP model can be mathematically defined as a tuple $< \mathcal{S}, \mathcal{A}, \mathcal{Z}, T, O, R >$ which contains a finite set of true states $\mathcal{S}$, a finite set of possible actions $\mathcal{A}$, the state transition probabilities $T : \mathcal{S} \times \mathcal{A} \rightarrow \Pi(\mathcal{S})$, a reward function depending on the state and the action just performed $R : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{R}$, a finite set of observations $\mathcal{Z}$ and a set of corresponding observation probabilities $O : \mathcal{S} \times \mathcal{A} \rightarrow \Pi(\mathcal{Z})$. A belief state is defined as the probability mass function over the current state, denoted as $b = (b(s_1), b(s_2), ...b(s_{|S|}))$, where $s_i \in \mathcal{S}, b(s_i) \geq 0$, and $\sum_{s_i \in \mathcal{S}} b(s_i) = 1$. A new belief state is re-estimated as $b^{t+1} = SE(b^t, a, z)$ which is further defined in Eqs.(1) and (2), given as

$$b^{t+1}(s_j) = \frac{O(s_j, a, z) \sum_{s_i \in \mathcal{S}} T(s_i, a, s_j) b^t(s_i)}{P(z|a, b^t)} \quad (1)$$

$$P(z|a, b^t) = \sum_{s_j \in \mathcal{S}} O(s_j, a, z) \sum_{s_i \in \mathcal{S}} T(s_i, a, s_j) b^t(s_i). \quad (2)$$

The reward function for action a performed at the belief state b is computed as $\rho(b, a) = \sum_{s_i \in \mathcal{S}} b(s_i) R(s_i, a)$. The belief data transition function becomes $\tau(b, a, b') = \sum_{z \in Z} p(b'|b, a, z) P(z|b, a)$ where $p(b'|b, a, z) = 1$ if $b' = SE(b, a, z)$, and 0 otherwise. To compute the optimal policy $\pi : \mathcal{R}^{|\mathcal{S}|} \rightarrow \mathcal{A}$, a value iteration function is typically involved, given as

$$V(b) = \max_a [\rho(b, a) + \gamma \sum_{b'} \tau(b, a, b') V(b')] \quad (3)$$

where γ is a discounting factor for the past history. In practice, it is common to have the optimal policy represented by a set of linear functions (so called α vectors) over the belief space, with the maximum "envelop" of the intersected α vectors forming the value function (Cassandra, 1998).

2.2. Sample-driven vs Value-directed Belief Compression

In the literature, there exist two main approaches for belief compression, namely *sample-driven* and *value-directed*.

The sample-driven approach (N. Roy and G. Gordon & Thrun, 2005) addresses POMDPs' curse of dimensionality problem by applying dimension reduction techniques like exponential principal component analysis (EPCA) so that the high-dimensional belief space can be characterized by a compact set of belief state basis vectors. While this approach has been shown to be effective in making POMDPs with sparse belief space more tractable, the non-linear projection makes the value function of the projected belief state space no longer piecewise-linear. Efficient algorithms which take advantage of the value function's PWLC property become inapplicable. While the policy can still be computed via a grid-based approximation (N. Roy and G. Gordon & Thrun, 2005), the implementation of the approximation is not straight-forward and yet introduces to it additional inaccuracy. Also, the sample-driven approach takes no consideration regarding how the value function in the compressed belief space is different from that in the original space.

The value-directed approach (Poupart & Boutilier, 2003) computes the minimal *Krylov* subspaces and thus the corresponding reward and state transition functions so that the value function will take same values for corresponding belief states in the original and compressed spaces. As the projection is linear, the value function after the projection is PWLC, and thus most of the existing algorithms for policy computation can be adopted. Computing the *Krylov* sub-space is however time-consuming as a large number of linear programming problems are to be solved and yet a high compression ratio cannot be guaranteed. A truncated *Krylov* iteration algorithm has also been introduced (Poupart & Boutilier, 2003) for obtaining a forcibly compressed POMDP by quickly stopping the *Krylov* iterations instead of deriving the complete set of belief basis vectors. This approach falls short as it provides no mechanism for exploring belief state samples at all.

3. A Value-directed Formulation with NMF Integrated

In this section, we review one of our recently proposed methodologies (Li et al., 2005a) which adopts non-negative matrix factorization (NMF) for the projection and integrates NMF into a value-directed framework.

3.1. Non-negative Matrix Factorization

With the objective to preserve the value function's PWLC property, we adopted non-negative matrix factorization (NMF) (Lee & Seung, 1999) for belief compression. NMF is a *linear* factorization technique which can guarantee all the data representations in the factored space and their reconstructed versions in the original data space to be *non-negative*. Given V

[1] All the POMDP benchmark problems can be found in http://www.pomdp.org/pomdp/examples/index.shtml

to be an $m \times n$ matrix with its columns being the observation vectors, one can approximate V using NMF so that $V \approx WH$ where W is an $m \times p$ matrix with its columns forming a set of p (normally $< m$) non-negative basis components and the matrix H are the corresponding coefficients. In other words, V is approximately represented by a weighted sum (H) of the basis components (W). W and H can be derived using multiplicative updating rules, given as

$$W_{ik} \leftarrow W_{ik} \frac{(VH^\top)_{ik}}{(WHH^\top)_{ik}} \quad (4)$$

$$H_{kj} \leftarrow H_{kj} \frac{(W^\top V)_{kj}}{(W^\top WH)_{kj}}. \quad (5)$$

Eqs.(5) and (4) alternate until W and H converge. It has been shown that the updating rules are in effect minimizing a Euclidean distance between V and WH. The computational complexity of NMF per iteration can easily be shown to be $O(nmp)$.

3.2. Proposed Formulation

Let B be a $n \times |S|$ matrix $[b_1|b_2|...|b_n]^\top$ representing the belief sample (obtained via simulation) where n is the size of the belief sample and b_i corresponds to a particular belief sample point. Also, let F be a $|S| \times l$ transformation matrix which factors B into the matrices F and $\widetilde{B}$ such that

$$B^\top \approx F^\top \widetilde{B}^\top \quad (6)$$

where each row of B equals $b \approx b^r = \widetilde{b}F$ and the dimension of $\widetilde{B}$ is $n \times l$. Here, $\widetilde{B}$ refers to the belief sample represented in the compressed space. As the main objective of deriving F is for dimension reduction, it is typical that $l << |S|$. One can compute Eq.(6) using NMF which minimizes the Euclidean distance between $B^\top$ and its reconstructed versions $F^\top \widetilde{B}^\top$. By equating the expected reward of a belief state b in the original belief space $V(b) = \sum_{s_i} b(s_i) \cdot R(s_i, a) = bR_{\cdot a} = \widetilde{b}FR_{\cdot a}$ and that for the compressed belief space $V(\widetilde{b}) = \widetilde{b}\widetilde{R}_{\cdot a}$, the reward function for the compressed POMDP becomes

$$\widetilde{R} = FR. \quad (7)$$

To derive the state-transition function in the compressed space, consider two different paths for computing the next belief state in the high-dimensional space. Given the current belief state in the original space, one path is to first apply the compression and then perform Bayesian updating in the compressed space. Another path is to first perform Bayesian updating in the original space and then perform the belief compression. One can then obtain

$$\begin{aligned} (\widetilde{b}^{t+1})^\top &= \widetilde{SE}(\widetilde{b}^t, a, z) \\ &= \widetilde{G}^{<a,z>}(\widetilde{b}^t)^\top \quad (8) \\ (b^{t+1})^\top &= F^\top(\widetilde{b}^{t+1})^\top \\ &= SE(b^t, a, z) \\ &= G^{<a,z>}(b^t)^\top \\ &= G^{<a,z>}F^\top(\widetilde{b}^t)^\top. \quad (9) \end{aligned}$$

Equating Eqs.(8) and (9) using Eq.(6), we obtain

$$F^\top \widetilde{G}^{<a,z>} = G^{<a,z>}F^\top. \quad (10)$$

If F and $\widetilde{G}$ that satisfy Eqs.(7) and (10) can be found, the compressed POMDP will be well-defined and can be readily solved. Note that the value iteration function is piecewise linear and thus commonly represented as sets of coefficient vectors (commonly called α vectors) which correspond to the hyperplanes forming the envelop of the value function. Getting $\widetilde{R}$ is straightforward but getting a good enough $\widetilde{G}$ is not. In a preliminary study, we have used pseudo inverse to obtain $\widetilde{G}$ but the quality of the policy obtained was bad. In Section 4, we will show how this limitation can be alleviated by introducing orthogonality into the NMF.

3.3. High-dimensional Policy Recovery

Once the formulation of the compressed POMDP is defined and solved, the policy obtained in principle could be used by the agent for taking actions rationally. However, updating the belief state in the compressed state turns out to be non-trivial. While NMF can guarantee elements of compressed belief states to be non-zero, it cannot guarantee their sum to be 1, which is assumed in Bayesian updating (Eq.(1)). One simple walk-around is to avoid this by recovering the high-dimensional α vectors from the low-dimensional ones so that the agent can act according to the recovered policy and update the belief state in the high dimensional belief space.

From the value-directed perspective, one can derive α vectors in the high-dimensional space such that $\alpha b^\top = \widetilde{\alpha}\widetilde{b}^\top$ holds[2]. As $b^\top = F^\top \widetilde{b}^\top$, we can easily obtain

$$\widetilde{\alpha} = \alpha F^\top. \quad (11)$$

The high-dimensional α vectors can thus be recovered if Eq.(11) can be solved.

4. Orthogonal NMF Belief Compression

4.1. Motivation

Recall that F is a $|S| \times l$ projection matrix. Usually we assume $|S| > l$ to achieve dimension reduction. So,

[2]Note that α and b are row vectors.

the equation $\widetilde{\alpha} = \alpha F^\top$ is under-constrained and has infinitely many solutions for α. In order to come up with a closed-form solution for α, we apply an additional constraint $F^\top F = I$ and Eq.(11) will become

$$\alpha = \widetilde{\alpha} F. \tag{12}$$

In addition, this can result in another closed-formed solution for $\widetilde{G}$ where Eq.(10) can be rewritten as

$$\widetilde{G}^{<a,z>} = F G^{<a,z>} F^\top. \tag{13}$$

The remaining question is how to enforce the orthogonality in F. We will call such a factorization O-NMF in the remaining sections.

4.2. Updating Rules of O-NMF

With the orthogonality constraint added, our compression problem becomes finding an F such that

$$B^\top = F^\top \widetilde{B}^\top \quad s.t. \quad F^\top F = I. \tag{14}$$

Using NMF's conventional notation scheme, it can be rewritten as as

$$V = WH, \quad s.t. \quad WW^\top = I. \tag{15}$$

As there are $|S| \times l$ variables for W and $|S|^2$ equations in $WW^\top = I$, the problem is over-constrained. Here, we propose the following updating rules (to be derived in the next section):

$$W_{ik} \leftarrow W_{ik} \sqrt{\frac{(VH^\top)_{ik}}{(WHH^\top + VH^\top W^\top W - WHH^\top W^\top W)_{ik}}} \tag{16}$$

$$H_{kj} \leftarrow H_{kj} \frac{(W^\top V)_{kj}}{(W^\top WH)_{kj}}. \tag{17}$$

which minimize $\|V - WH\|_F^2$ as well as $\|WW^\top - I\|_F^2$.[3]

4.3. Derivation of Updating Rules

We adopt an auxiliary function-based method (Lee & Seung, 2001) for constructing NMF updating rules. The general idea is that if $L(h)$ is the objective function to be minimized in NMF and $Z(h, h')$ is an auxiliary function satisfying the following conditions

$$Z(h, h') \geq L(h), \quad Z(h, h) = L(h), \tag{18}$$

for any h and h', the NMF updating rule can be constructed by defining

$$h^{(t+1)} = \arg\min_h Z(h, h^{(t)}). \tag{19}$$

It is obvious to see that $L(h^{(t)}) = Z(h^{(t)}, h^{(t)}) \geq Z(h^{(t+1)}, h^{(t)}) \geq L(h^{(t+1)})$ according to Eqs.(18) and (19). Thus $L(h^{(t)})$ is non-increasing and thus the updating rule's convergence can be guaranteed.

[3] $\|X\|_F$ refers to the Frobenius norm of the matrix X.

So to derive the updating rules for our constrained problem, the key here is to find an appropriate auxiliary function. The objective function for NMF (Eq.(15)) is

$$L = \|V - WH\|_F^2. \tag{20}$$

The Lagrangian multiplier method can be used for the constrained minimization. (Ding et al., 2006). The induced lagrangian function becomes

$$L_\lambda = \|V - WH\|_F^2 + Tr[\lambda(WW^\top - I)] \tag{21}$$

which in turn can be rewritten as

$$L(W) = Tr[-2W^\top VH^\top + WHH^\top W^\top + \lambda WW^\top] \tag{22}$$

where the constants $Tr[V^\top V]$ and $Tr[\lambda]$ are ignored. It can then be shown that the following function

$$\begin{aligned} Z(W, W') = &- \textstyle\sum_{ik} 2(VH^\top)_{ik} W'_{ik} (1 + \log \frac{W_{ik}}{W'_{ik}}) \\ &+ \textstyle\sum_{ik} \frac{W'(HH^\top)W^2_{ik}}{W'_{ik}} + \sum_{ik} \frac{\lambda W' W^2_{ik}}{W'_{ik}} \end{aligned} \tag{23}$$

is an auxiliary function of $L(W)$. The proof can be found in Appendix A.

Based on Eq.(19), we compute $\arg\min_W Z(W, W')$ by setting

$$\frac{\partial Z(W, W')}{\partial W_{ik}} = -2(VH^\top)_{ik} \frac{W'_{ik}}{W_{ik}} + 2\frac{W'HH^\top W_{ik}}{W'_{ik}} + 2\frac{\lambda W' W_{ik}}{W'_{ik}}. \tag{24}$$

to 0 and obtain

$$W_{ik} = W'_{ik} \sqrt{\frac{(VH^\top)_{ik}}{(W'HH^\top + \lambda W)_{ik}}}. \tag{25}$$

To determine the value of λ, we take the gradient of $L(W)$ again and obtain

$$\lambda = VH^\top W^\top - WHH^\top W^\top. \tag{26}$$

Based on Eq.(19), W's updating rule becomes

$$W_{ik} = W'_{ik} \sqrt{\frac{(VH^\top)_{ik}}{(WHH^\top + VH^\top W^\top W - WHH^\top W^\top W)_{ik}}}. \tag{27}$$

During the NMF iterations, we require $\lambda \geq 0$ due to the square root in Eq.(25). For those $VH^\top W^\top - WHH^\top W^\top$ which give negative values, we set them to 0 to guarantee the denominator in Eq.(27) to be positive and thus the nonnegative property can be maintained. Given a particular W, we can update H using the standard updating rule as Eq.(5). Even with the λ truncation, we found in our experiments that the convergence of the NMF was still stable and the reconstruction loss decreases gradually with no fluctuations. However, this truncation does result in some

increase in the reconstruction error and deviation from the orthogonality constraint, if the initialization of W and H are not carefully done.

To obtain a good initialization of W which can satisfy the constraint $WW^\top = I$, we developed a two-step NMF procedure. Step 1 is to obtain $V = WH'$ using the conventional NMF, which is to be followed by Step 2 which solves $H' = W^\top V$ using a different version of orthogonal NMF proposed in (Ding et al., 2006) where the orthogonality constraint is $W^\top W = I$. By such a setting, the orthogonality constraint $(W^\top)^\top W^\top = WW^\top = I$ is set forth. We have tested the POMDP derived directly based on this initialization step. The reconstruction error was large but the orthogonality property of $WW^\top = I$ was well satisfied. By using it as the initialization for our O-NMF updating rule in Eq.(27), we noticed further substantial reduction in the loss $\|V - WH\|_F^2$ and at the same time good orthogonality achieved for W. Algorithm 1 summarizes the main steps of our O-NMF. The computational complexity of O-NMF can be shown to be different from that of NMF by a constant factor.

With the O-NMF in place, we can then easily compute the POMDP formulation for the compressed space. Since O-NMF does a linear projection, the compressed POMDP can be solved using any existing POMDP algorithm. Algorithm 2 gives an overall picture of our proposed methodology.

Algorithm 1 O-NMF

Input: data V, reduced dimension p
Output: W, H
/ initialize W and H */*
Set W and H randomly.
for $i = 1$ **to** $iterNum$ **do**
  $H_{kj} \leftarrow H_{kj} \frac{(W^\top V)_{kj}}{(W^\top W H^\top)_{kj}}$
  $(W^\top)_{ik} \leftarrow (W^\top)_{ik} \sqrt{\frac{(HV)_{ik}}{(W^\top W H V)_{ik}}}$;
end for
/ update W, H using O-NMF updating rule */*
while $\delta > \epsilon$ **do**
  $\delta = \|V - WH\|_F^2$
  $\lambda = VH^\top W^\top - WHH^\top W^\top$
  set -ve λ_{ik} to 0
  $H_{kj} = H_{kj}(W^\top V)_{kj}/(W^\top W H^\top)_{kj}$
  $W_{ik} = W_{ik}\sqrt{(VH^\top)_{ik}/(WHH^\top + \lambda W)_{ik}}$
end while
return

5. Implementation Details and Performance Evaluation

In this section, we first explain the algorithm we adopted for computing the policy and a trick we used for the value function initialization. Then, we present performance evaluation results to demonstrate empirically the effectiveness and efficiency gained by the proposed O-NMF belief compression.

Algorithm 2 O-NMF-based Belief compression

Input: original POMDP problem S, reduced dimension l
Output: α vectors (policy) for the problem
Sample beliefs and store them as row vectors in a matrix B.
Invoke O-NMF to solve $B^\top = F^\top \widetilde{B}$
$\widetilde{R} = FR$
$\widetilde{G}^{<a,z>} = FG^{<a,z>}F^\top$
Compute optimal policy for POMDP defined by $\widetilde{R}$ and $\widetilde{G}^{<a,z>}$ and get the low-dimensional policy($\widetilde{\alpha}$ vectors)
$\alpha = \widetilde{\alpha}F$
return

5.1. Computing Policy Using Point Based Value Iteration

In our experiments, Perseus — an efficient randomized point-based approximate value iteration algorithm (Spaan & Vlassis, 2005) was adopted for computing the policy. Combining Perseus into our framework is straightforward, except that Perseus requires a *backup* belief set for reducing the number of α vectors to be stored. We used the same belief sample created using the trajectory-based approach for O-NMF to be also the backup belief set.

5.2. Tighter Initial bound for α Vector

Perseus starts with a single α vector initialized with a lower bound of the value function. In fact, a good lower bound can effectively speed up the convergence of the value iteration. For some of the problems we tested, we used a tighter lower bound based on the one-step immediate reward

$$\alpha^a(s) = R(s, a) \tag{28}$$

instead of conventional initialization

$$\alpha(s) = \frac{\min_{s,a} R(s, a)}{1 - \gamma} \tag{29}$$

used in Perseus (Pineau et al., 2003). The rationale is that for problems with $\min_{s,a} R(s, a) < 0$, Eq.(29) will further lower the initial lower bound of α vectors and thus need more value iteration for convergence. In Fig.1, the bottom left subfigure shows the α vector initialization using Eq.(28) and the upper left shows

the conventional initialization using Eq.(29). Initialized as in the bottom left, the value function converged to the optimal policy as shown in the bottom right in only 5 steps. But the traditional initialization (upper left) converges to the optimal policy (upper right) by taking 31 steps instead. In our experiments, we have tested the suggested initialization method, the conventional one as well as yet another recently proposed one (Smith & Simmons, 2005). We found that for some unsolvable problems for Perseus, our suggested initialization can make them becomes solvable but not the other two as mentioned.

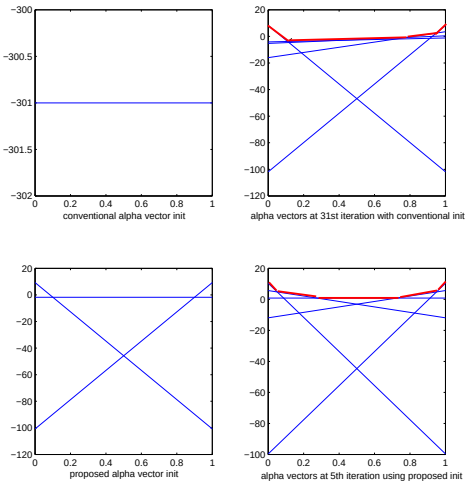

Figure 1. Comparison of two α vector initializations.

5.3. Effectiveness in Belief Reconstruction

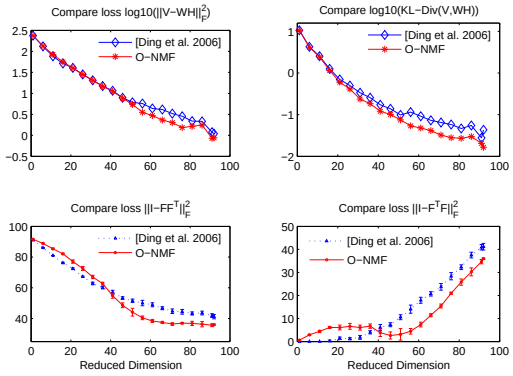

Figure 2. Reconstruction accuracy comparison between O-NMF and [Ding et al. 2006] based on 500 belief samples obtained from the Hallway2 problem.

As mentioned in Section 4.3, our proposed version of orthogonal NMF is different from that proposed in (Ding et al., 2006) regarding the orthogonality constraint and Ding *et.al*'s version of orthogonal NMF is not applicable for our case. However, we still conducted reconstruction accuracy comparison to see if there is significant difference given the change in the orthogonality constraint. To our surprise, the empirical results we obtained show that our proposed version of O-NMF performed better than Ding *et.al*'s version regarding belief reconstruction and orthogonality satisfaction. For example, for the Hallway2 problem, our method achieved (a) lower belief reconstruction errors measured in terms of matrix norm and KL-divergence, and (b) lower orthogonality deviations measured in terms of $||W^\top W - I||_F^2$ and $||WW^\top - I||_F^2$ as shown in Fig. 2. The performance difference was more obvious when the reduced dimension was 40 or above.

5.4. Policy Quality and Computational Efficiency

For performance comparison, we implemented the proposed O-NMF belief compression as well as EPCA-based belief compression (N. Roy and G. Gordon & Thrun, 2005) and truncated Krylov compression (Poupart & Boutilier, 2003) using Matlab 7.0. We ran all our experiments on a machine with a Dual Xeon CPU 3.06GHz and the memory size of 2 GB. Full matrices were used to store data in Matlab[4]. Due to the non-linear nature of EPCA, we implemented also a particular grid-based approximation for policy computation as suggested in (Poupart & Boutilier, 2003). However, our previous work (Li et al., 2005b) showed that the policy quality we obtained using EPCA-based compression for the Hallway2 problem was far from the optimal one. Also, its performance on benchmark problems like the Hallway2 problem were not reported in (Poupart & Boutilier, 2003). Therefore, in this paper, we only compare in detail our O-NMF based belief compression and truncated Krylov compression.

To help understand the behavior of the proposed O-NMF compression, Fig.3 shows the detailed performance in terms of belief reconstruction error and average reward for solving the Hallway problem at different reduced dimensions. The upper subfigure shows the average KL-divergence between the original 500 belief samples and the reconstructed versions derived from their compressed counterparts. The middle one shows the orthogonality derivation. The lower subfigure shows the average reward. It can be observed that accurate reconstruction and good orthogonality should both be ensured before good quality policies can be obtained. For this 60-dimensional problem, our proposed approach could still give a nearly optimal policy when the dimension was reduced to 37 and could achieve an

[4]Note that using sparse matrix data structure can further speed up O-NMF and the value iteration step.

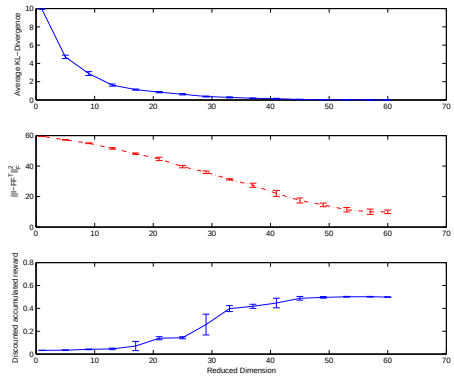

Figure 3. Performance of O-NMF belief compression for the Hallway problem compressed to different extents.

essentially lossless low-dimensional policy when the dimension was 45.

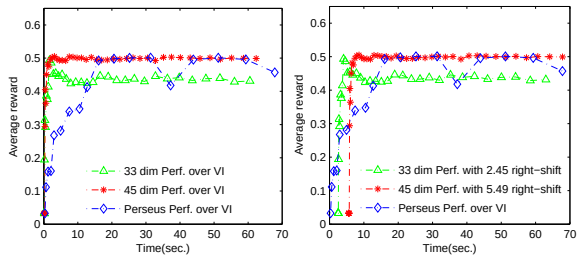

Figure 4. Detailed performance comparison of the proposed methodology over time within Perseus.

Fig.4 gives a close look at the change of average award for the policies computed over time for the Hallway problem using only Perseus and Perseus with O-NMF belief compression where the dimension was reduced to 33 and 45. It can be observed from the left subfigure that Perseus with O-NMF converged rapidly to the optimal policy. The right subfigure is essentially the same as that of the left one, except that we have taken into account also the O-NMF compression overhead (2.45s and 5.49s corresponding to the cases of reduced dimensions being 33 and 45 respectively) and right-shifted the curves accordingly. For the Hallway problem, the overhead is obviously insignificant.

We have also conducted a more rigorous performance comparison between O-NMF compression and truncated Krylov compression. Table 1 tabulates the results obtained in terms of (a) running time (with breakdown of the time needed by O-NMF and Perseus) and (b) the policy's average award. A number of benchmark problems have been tested. It can be observed that for the problems *Tiger-grid*, *Hallway*, *Hallway2*, and *RockSample*, the cases with O-NMF can give policies of quality almost the same as that without it at a reasonable reduced dimension. Also, the computational speedup brought by the O-NMF was quite significant for most of the cases, except for Hallway2. For the Pentagon problem, both Perseus and Perseus with O-NMF cannot solve it. However, it is interesting to point out that we managed to solve the Pentagon problem using Perseus by adopting the initialization discussed in Section 5.2.

Table 1. Performance comparison for different problems. (T_p - policy computing time; T_c - compression time)

Problem (states/actions/obs.)	Reward	Time(sec.) $T_p + T_c$	Reduced Dim.
Tiger-grid(*500 samples*) (36s 5a 17o)	±0.014		
Perseus	0.63	104	*N/A*
Perseus+Tunc.Kry.	-ve	–	–
Perseus+O-NMF	0.59	18.14+0.2	28
Perseus+O-NMF	0.65	10.26+0.2	30
Hallway(*500 samples*) (60s 5a 21o)	±0.005		
Perseus	0.50	53	*N/A*
Perseus+Tunc.Kry.	0.44	5.6 + 41.27	33
Perseus+Tunc.Kry.	0.50	13.78 + 202	48
Perseus+O-NMF	0.43	1.65+2.45	33
Perseus+O-NMF	0.50	2.46+5.49	45
Hallway2(*500 samples*) (92s 5a 17o)	±0.014		
Perseus	0.31	69.96	*N/A*
Perseus+Tunc.Kry.	0.29	36 + 52.7	48
Perseus+O-NMF	0.29	49+8.398	60
Pentagon(*1000 samples*) (212s 4a 28o)	±0.0015		
Perseus	-	-	*N/A*
Perseus+T.Init	0.8132	691.3	*N/A*
RockSample(*60 samples*) (257s 9a 2o)	±1.22		
Perseus	7.96	35	*N/A*
Perseus+Tunc.Kry.	-	-	11
Perseus+O-NMF	6.53	2.7+0.3	11

Comparing our proposed approach with truncated Krylov compression (as shown in Table 1), truncated Krylov compression got negative reward values over all reduced dimensions for the Tiger-grid problem, and comparable results as ours for Hallway and Hallway2 problems but requiring a much longer compression time (202s and 52.7s respectively). For the RockSample problem, it achieved only a reward value of 2.27 at dimension 21 and also failed to solve the *Pentagon* problem. The proposed O-NMF based belief compression obviously outperforms truncated Krylov compression regarding its effectiveness in reducing the POMDP's computational complexity.

6. Conclusion

This paper describes a novel orthogonal NMF-based POMDP compression which on one hand explores the belief space's sparsity for dimension reduction and at the same time can efficiently compute the compressed POMDP formulation which is also value-directed. The proposed approach has been demonstrated to be effective in improving the tractability of POMDP based on a set of benchmark problems.

This work can be further extended at least in the two directions. (1) O-NMF needs a significant number belief sample points before it is accurate enough. The computational overhead however will increase with the belief sample set. We are currently investigating how to carefully control the complexity of this overhead portion. (2) The degree to which a belief space can be compressed could be intrinsically limited for some problems. As demonstrated in (Li et al., 2005a), it would be interesting to investigate the possibility for the belief space to be clustered for more effective per-cluster belief compression.

Acknowledgments

We would like to thank the anonymous reviewers for their useful and insightful comments. This work has been partially supported by RGC Central Allocation Group Research Grant (HKBU 2/03/C) and HKBU Faculty Research Grant (FRG/05-06/II-81).

References

Cassandra, A. (1998). *Exact and approximate algorithms for partially observable Markov decision processes*. Doctoral dissertation, Brown University.

Ding, C., Li, T., Peng, W., & Park, H. (2006). Orthogonal nonnegative matrix t-factorizations for clustering. *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Philadelphia, PA, USA* (pp. 126–135).

Lee, D. D., & Seung, H. S. (1999). Learning the parts of objects by non-negative matrix factorization. *Nature, 401*, 788–791.

Lee, D. D., & Seung, H. S. (2001). Algorithms for non-negative matrix factorization. In *Neural information processing systems 13*, 556–562. MIT Press.

Li, X., Cheung, W. K., & Liu, J. (2005a). Decomposing large scale POMDP via belief state analysis. *Proceedings of 2005 IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT'05)* (pp. 428–434). Compiegne, France.

Li, X., Cheung, W. K., & Liu, J. (2005b). Towards solving large-scale POMDP problems via spatio-temporal belief state clustering. *Proceedings of IJCAI-05 Workshop on Reasoning with Uncertainty in Robotics (RUR'05)*. Edinburgh, Scotland.

N. Roy and G. Gordon, & Thrun, S. (2005). Finding approximate POMDP solutions through belief compressions. *Journal of Artificial Intelligence Research, 23*, 1–40.

Pineau, J., Gordon, G., & Thrun, S. (2003). Point-based value iteration: An anytime algorithm for POMDPs. *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI-03)*.

Poupart, P., & Boutilier, C. (2003). Value-directed compression of POMDPs. In *Advances in Neural Information Processing Systems 15*, 1547–1554. Cambridge, MA: MIT Press.

Smith, T., & Simmons, R. (2005). Point-based POMDP algorithms: Improved analysis and implementation. *Proceedings of the 21th Annual Conference on Uncertainty in Artificial Intelligence (UAI-05)* (pp. 542–55). Arlington, Virginia: AUAI Press.

Spaan, M. T. J., & Vlassis, N. (2005). Perseus: Randomized point-based value iteration for POMDPs. *Journal of Artificial Intelligence Research, 24*, 195–220.

A. Detailed Proof of the Auxiliary Function

By referring to Eqs. (22) and (23), it is obvious to see that $Z(W, W') = L(W)$ when $W' = W$. To show that the inequality $Z(W, W') \geq L(W)$ also holds, the first part of the proof is to show that the second and third terms in $Z(W, W')$ are always bigger than the corresponding terms in $L(W)$ respectively. First, we have the following property (Ding et al., 2006)

$$\sum_{i=1}^{n}\sum_{p=1}^{k}\frac{(AS'B)_{ip}S_{ip}^2}{S'_{ip}} \geq Tr(S^\top ASB)$$

where $A \in \mathbb{R}_+^{n\times n}$, $B \in \mathbb{R}_+^{k\times k}$, $S \in \mathbb{R}_+^{n\times k}$, $S' \in \mathbb{R}_+^{n\times k}$, and A and B are symmetric. Since $Tr(S^\top ASB) = Tr(ASBS^\top)$, we can take $A = I, S = W, B = H^\top H$ and $A = \lambda, S = W, B = I$ and the first part of the proof follows. The remaining part corresponds to the first term. We can prove that the first term in $Z(W, W')$ is again always bigger than that in $L(W)$ by setting $z = W_{ik}/W'_{ik}$ in the following equality

$$-(1 + \log(z)) \geq -z, \forall z > 0.$$

This completes the proof.