Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence (AAAI-10)

Compressing POMDPs Using Locality Preserving Non-Negative Matrix Factorization

Georgios Theocharous
Intel Corporation
Intel Labs
Santa Clara

Sridhar Mahadevan
Computer Science Department
University of Massachusetts
Amherst

Abstract

Partially Observable Markov Decision Processes (POMDPs) are a well-established and rigorous framework for sequential decision-making under uncertainty. POMDPs are well-known to be intractable to solve exactly, and there has been significant work on finding tractable approximation methods. One well-studied approach is to find a compression of the original POMDP by projecting the belief states to a lower-dimensional space. We present a novel dimensionality reduction method for POMDPs based on locality preserving non-negative matrix factorization. Unlike previous approaches, such as Krylov compression and regular non-negative matrix factorization, our approach preserves the local geometry of the belief space manifold. We present results on standard benchmark POMDPs showing improved performance over previously explored compression algorithms for POMDPs.

Introduction

Partially Observable Markov Decision Processes (POMDPs) provide a rigorous mathematical framework for sequential decision making under uncertainty (Smallwood and Sondik 1973). A POMDP formalizes how agents can act optimally in the presence of noisy observations and stochastic actions. Due to partial observability, an agent can use *belief states* or probability distributions over the true states S of the POMDP to represent the history of past observations and actions. Belief states constitute a sufficient statistic, in that they are Markov and allow the agent to predict next belief states from the current one. This property allows formulating the problem of solving a POMDP as an $|S| - 1$ dimensional continuous belief state Markov Decision Process. The dimensionality of the belief space makes the problem computationally intractable. The computational complexity of optimally solving a POMDP in the finite-horizon setting with t steps lookahead can be 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. The space complexity of representing a value function gets worse as the belief space dimensionality increases. There have also been results showing that some POMDPs can be intrinsically hard to approximate to within a constant factor (Lusena, Goldsmith, and Mundhenk 2001). Despite this result, there has been significant progress on approximation algorithms for solving POMDPs, such as point-based value iteration (PBVI) (Pineau, Gordon, and Thrun 2006; Spaan and Vlassis 2005). There has also been theoretical work to explain the apparent successes of PBVI-based methods (Hsu, Lee, and Rong 2007) in terms of *covering numbers*, the number of ϵ-sized balls needed to cover the belief space. Essentially, approximately optimal solutions to POMDPs can be computed in time polynomial in the covering number.

Dimensionality reduction in POMDPs can be achieved by belief compression, which projects the high-dimensional belief space to a lower-dimensional one, thereby reducing the policy computation time, while taking care to not significantly degrade the quality of the policy. When dimensionality reduction is done in a linear fashion, then one can produce a low dimensional POMDP model and use most existing policy computation algorithms for POMDPs. Two approaches in the literature to linear dimensionality reduction include *Krylov* bases (Poupart and Boutilier 2002) and orthogonal non-negative matrix factorization (ONMF) (Li et al. 2007). While the first approach computes a low dimensional subspace by solving a set of linear programs, the second approach uses non-negative matrix factorization over a sample of belief points.

In this paper, we extend the NMF approach with an additional *locality preserving* constraint, which requires that if two points are geometrically close in the original space they should also be geometrically close in the reduced space. Our idea to preserve locality is motivated by the fact that in the finite-horizon setting, POMDP value functions are piecewise-linear and convex, whereas in the infinite-horizon setting, POMDP value functions are convex. Indeed, an important property of POMDP value functions is that they satisfy a *Lipschitz* continuity property over the belief space: specifically, if two belief states are within δ of each other (in L_1 distance), then the optimal value function changes by at most δ times a constant factor (which is $\frac{R_{max}}{1-\gamma}$) (Hsu, Lee, and Rong 2007). Therefore, a reduction that preserves the locality of points should be able to better capture the original value function on the uncompressed space. To implement our approach we apply a newly proposed algorithm called locality preserving non-negative matrix factorization (LPNMF) (Cai et al. 2009). This approach uses a graph Laplacian on the sampled (belief) space as a regularizer to enforce locality preservation of the embedding. Nonlinear dimensionality reduction methods based on the graph Laplacian have been widely explored in machine learning (Belkin and Niyogi 2003), but this approach has not been studied in the context of POMDPs, to the best of our knowledge. We validate LPNMF on a set of benchmark problems and demonstrate significantly better compression than the previously studied ONMF approach.

Here is a roadmap to the remainder of the paper. First, we give an overview of POMDPs and describe linear dimensionality reduction methods. Subsequently, we discuss non-negative matrix factorization methods, and specifically describe the locality preserving NMF method. We then describe how this approach is used to solve POMDPs. Finally we present our experimental results, and conclude with a discussion of future work.

Review of POMDPs

Partially observable Markov decision processes (POMDPs) provide a rigorous mathematical framework for planning under uncertainty in both actions and observations (Kaelbling, Littman, and Cassandra 1998; Smallwood and Sondik 1973; 1978). A POMDP is defined as a six tuple $\langle S, A, Z, T, O, R\rangle$, where S is a set of states, A is a set of actions, Z is a set of observations, T is a stochastic transition function, O is the stochastic observation function, and R the reward function. At each discrete time step, the environment is in some state $s \in S$; an agent takes action $a \in A$ from which it receives a reward $R(s, a)$. As a consequence, the environment transitions to state $\acute{s} \in S$ with probability $P(\acute{s}|s, a) = T(s, a, \acute{s})$, and the agent observes $z \in Z$ with probability $P(z|\acute{s}, a) = O(\acute{s}, a, z)$. The goal of POMDP planning is to find a policy that, based upon the previous sequence of actions and observations, chooses actions that maximize the expected discounted sum of rewards $\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t)$, where $\gamma \in [0, 1)$ is a discount factor.

In a POMDP the states are not observed directly. Instead the agent maintains an internal belief state b, defined as the probability distribution over states given past actions and observations. It is well known that the belief state is a sufficient statistic for a given history of actions and observations (Smallwood and Sondik 1973), and it is updated at each time step by incorporating the latest action and observation via Bayes rule:

$$b_a^z(\acute{s}) = \frac{O(\acute{s}, a, z)\sum_{s\in S} T(s, a, \acute{s})b(s)}{\sum_{\acute{s}\in S} O(\acute{s}, a, z)\sum_{s\in S} T(s, a, \acute{s})b(s)}; \quad (1)$$

where b_a^z denotes the belief state updated from b by taking action a and observing z.

The dynamic behavior of the belief state is itself a discrete-time continuous-state Markov process (Smallwood and Sondik 1973), and a POMDP can be recast as a completely observable MDP with a $(|S| - 1)$-dimensional continuous state space. Based on these properties, several exact algorithms (Cassandra, Littman, and Zhang 1997; Kaelbling, Littman, and Cassandra 1998) have been developed. However, because of the exponential worst-case complexity (Lusena, Goldsmith, and Mundhenk 2001), these algorithms typically are limited to solving small problems.

Point-based Value Iteration

The poor scalability of exact algorithms has led to the development of an approximate solution method called Point Based Value Iteration (PBVI) (Pineau, Gordon, and Thrun 2006; Spaan and Vlassis 2005). Unlike exact algorithms, which plan over the entire belief simplex, PBVI algorithms approximate the exact solution by planning only over a finite set of belief points B. They utilize the fact that most practical POMDP problems assume an initial belief b_0, and concentrate planning resources on regions of the simplex that are reachable from b_0. Based on this idea, Pineau et al. (2006) proposed a PBVI algorithm that first collects a finite set of belief points B by forward simulating the POMDP model. The algorithm then computes over those belief states a set Γ of α vectors that represent the POMDP solution. Figure 1 describes how the point-based backup operation computes an α vector for every belief state b. During execution, for a given belief state b, a POMDP agent chooses the action a such that $a = \arg\max_{\alpha_a} \alpha_a \cdot b$.

Inputs:
Γ, current set of α vectors
B, set of sampled belief states
Algorithm:
for each $b \in B$
$\alpha_a^z = \arg\max_{\alpha\in\Gamma} \alpha \cdot b_a^z$, for every $a \in A$, $z \in Z$
$\alpha_a(s) = R(s, a) + \gamma \sum_{z,\acute{s}} T(s, a, \acute{s})O(\acute{s}, a, z)\alpha_a^z(\acute{s})$
$\acute{\alpha} = \arg\max_{\alpha_a} \alpha_a \cdot b$
if $\acute{\alpha} \notin \acute{\Gamma}$, **then** $\acute{\Gamma} \leftarrow \acute{\Gamma} + \acute{\alpha}$, **end**
end
Output:
$\acute{\Gamma}$, new set of alpha vectors

Figure 1: Point-based backup of the PBVI algorithm

Linear Dimensionality Reduction in POMDPs

Coupled with the idea of approximately solving a POMDP using a point-based method, we can further simplify the problem of solving a POMDP by seeking to compress the belief space reachable from the initial state. Thus, approximation methods such as PBVI can be accelerated since their complexity depends on the belief space dimensionality of the underlying POMDP model. An important constraint in dimensionality reduction is to focus the power of approximation on high-value states, that is to construct a *value-directed compression* method (Poupart and Boutilier 2002). A linear variant of value-directed compression is defined by a linear transformation matrix F of size $d \times l$, where d is the original dimensionality of the POMDP belief space, and l is the reduced one.

In lossless dimensionally reduction, the value function of a policy at any state in the reduced space should be equal to the value of the corresponding state in the uncompressed space. For lossless linear dimensionality reduction, the following two equations need to be satisfied (Poupart and Boutilier 2002; Li et al. 2007):

$$R = F \cdot \tilde{R} \tag{2}$$

$$G^{a,z} \cdot F = F \cdot \tilde{G}^{a,z} \quad \forall a, z, \tag{3}$$

where the transition function G is defined for every action a and observation z as:

$$G^{a,z}_{i,j} = T(s_i, a, s_j)O(s_j, a, z).$$

Defining $F^{\dagger}$ as the pseudo inverse of F, we can use Equations 2 and 3 to compute the compressed reward and transition functions as:

$$\tilde{R} = F^{\dagger} \cdot R \tag{4}$$

$$\tilde{G} = F^{\dagger} \cdot G \cdot F. \tag{5}$$

When these equations are satisfied, F provides a new basis that provides a compressed representation of the value function. The columns of F span a subspace containing the reward function, which is invariant with respect to the transition matrix. These properties in effect generate two paths for simulating the POMDP, as shown in Figure 2: first compress the belief state, and then compute the next belief state, or alternatively, compute the next belief state and then compress it.

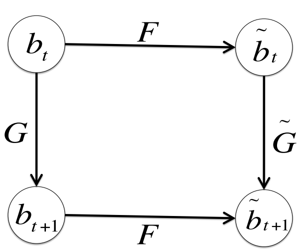

Figure 2: The figure shows two different paths for getting to an unnormalized belief state $\tilde{b}_{t+1}$ from an unnormalized belief state b_t.

Once a linear compression matrix has been found, the POMDP can be solved by running the point-based value iteration algorithm in Figure 1 on the compressed POMDP. The belief set $\tilde{B}$ to be used with the algorithm is projected from the original space as $\tilde{B} = B \cdot F$. Once the reduced POMDP is solved and an $\tilde{\alpha}$ solution is computed, we can execute the policy in the original POMDP by choosing actions such as $a = \arg\max_{\tilde{\alpha}_a} \tilde{\alpha}_a \cdot b \cdot F$. Poupart and Boutilier (2002) show that such an F can be thought of as the minimal *Krylov* subspace, which is invariant with respect to F and contains R. Computing the *Krylov* subspace that achieves a specified amount of lossy compression can be time consuming, involving a large number of linear programming problems.

An alternative to computing the minimal Krylov space was proposed in (Li et al. 2007), using a novel orthogonal non-negative matrix factorization algorithm to compute $F^{\dagger}$. Non-negative matrix factorization (NMF) is a technique to factorize a matrix X into two non-negative matrices, U and V, such that $X \approx U \cdot V^T$ (Lee and Seung 2001). The factorization can either be done to minimize the Euclidean distance between the points in the reconstructed and original space or to minimize the KL-divergence.

The ONMF approach described in (Li et al. 2007) first samples a set of n belief points and stores them in a matrix B of dimensions $n \times d$. Then, the ONMF algorithm is used to factor

$$B^T = (F^{\dagger})^T \cdot \tilde{B}^T. \tag{6}$$

This directly gives the pseudo inverse $F^{\dagger}$, which can be used to estimate $\tilde{R}$ in Equation 4. In addition, the algorithm enforces that the matrix is orthogonal, such that $(F^{\dagger})^T \cdot F^{\dagger} = I$. Having an orthogonal reduction makes it trivial to compute $F = (F^{\dagger})^T$.

Review of Non-negative Matrix Factorization

Non-negative matrix factorization (Lee and Seung 1999) is a dimensionality reduction method that decomposes data matrices whose elements are non-negative into a product of lower-rank non-negative matrices. This type of decomposition provides an intuitively more meaningful "parts-of" decomposition compared with other approaches that generate decompositions with negatively valued elements. In particular, since the space of belief states consists of non-negative vectors, NNMF seems an appropriate method to use. Given a data matrix $X = [x_{i,j}] \in \Re^{m \times n}$, where there are n points of dimensionality m, NMF aims to find two non-negative matrices $U = [u_{i,k}] \in \Re^{m \times t}$ and $V = [v_{j,k}] \in \Re^{n \times t}$ which minimize the following objective function:

$$\mathcal{O} = \sum_{i=1}^{m} \sum_{j=1}^{n} \left(x_{i,j} \log \frac{x_{i,j}}{y_{i,j}} - x_{i,j} + y_{i,j} \right); \tag{7}$$

where $Y = [y_{i,j}] = U \cdot V^T$. The above objective function is lower bounded by zero, and vanishes if and only if $X = Y$. Also it is convex in U only or V only, but not both variables together. Therefore it is difficult to design an efficient algorithm to find the global minimum of $\mathcal{O}$. Lee and Seung (Lee and Seung 2001) proved that the iterative updating of Equations 8 and 9 converges at a local minimum of the objective function:

$$u_{i,k} \leftarrow u_{i,k} \frac{\sum_j (x_{i,j} v_{j,k}) / \sum_k (u_{i,k} v_{j,k})}{\sum_j v_{j,k}} \tag{8}$$

$$v_{i,j} \leftarrow v_{j,k} \frac{\sum_j (x_{i,j} v_{i,k}) / \sum_k (u_{i,k} v_{j,k})}{\sum_j u_{i,k}}. \tag{9}$$

Locality Preserving NMF

To preserve the Lipschitz continuity property of POMDP value functions over the belief space, belief states that are close to each other should ideally be embedded in the lower-dimensional space close to each other as well. Although NMF by itself does not enforce such a constraint, it is possible to design a hybrid algorithm that combines the idea of non-negative matrix factorization and locality preserving projections (Cai et al. 2009). In the standard NMF the goal is to find a basis that is optimized for the linear approximation of the data. In the locality preserving NMF (LPNMF) approach, the optimization ensures that if two points are geometrically close in the original space, then they are also geometrically close in the projected space. Specifically, given a data matrix $X = [x_{i,j}] \in \Re^{m\times n}$, LPNMF aims to find two non-negative matrices $U = [u_{i,k}] \in \Re^{m\times t}$ and $V = [v_{j,k}] \in \Re^{n\times t}$ which minimize the following objective function:

$$\mathcal{O} = \sum_{i=1}^{m}\sum_{j=1}^{n}\left(x_{i,j}\log\frac{x_{i,j}}{y_{i,j}} - x_{i,j} + y_{i,j}\right) + \lambda\mathcal{R};$$

where λ is the regularization parameter. While the first part is the standard NMF optimization objective for minimizing KL-divergence between the original points X and reconstructed points Y, the $\mathcal{R}$ term is a constraint to ensure that geometrical locality holds over the reduced points in V. Specifically:

$$\mathcal{R} = \frac{1}{2}\sum_{j,s=1}^{n}\sum_{k=1}^{t}\left(v_{j,k}\log\frac{v_{j,k}}{v_{s,k}} + v_{s,k}\log\frac{v_{s,k}}{v_{j,k}}\right)W_{j,s};$$

where W is the weight matrix between the points in the original space X. Intuitively, minimizing $\mathcal{R}$ means that two points which are close together in the original space (i.e. $W_{j,s}$ is large), they will also be close in the reduced space. The following multiplicative algorithm can be used to minimize $\mathcal{O}$ and estimate the matrices U and V:

$$u_{i,k} \leftarrow u_{i,k}\frac{\sum_j (x_{i,j}v_{j,k})/\sum_k(u_{i,k}v_{j,k})}{\sum_j v_{j,k}} \tag{10}$$

$$v_k \leftarrow \left[\sum_i u_{i,k}I + \lambda L\right]^{-1}\cdot \begin{bmatrix} v_{1,k}\sum_i(x_{i,1}u_{i,k})/\sum_k(u_{i,k}v_{1,k}) \\ v_{2,k}\sum_i(x_{i,2}u_{i,k})/\sum_k(u_{i,k}v_{2,k}) \\ \vdots \\ v_{n,k}\sum_i(x_{i,n}u_{i,k})/\sum_k(u_{i,k}v_{n,k}) \end{bmatrix}; \tag{11}$$

where v_k is the k-th column of V and I is an $n \times n$ identity matrix. The matrix L is the graph Laplacian (Chung 1997).

Locality Preserving NMF for POMDPs

The algorithmic steps of the overall algorithm to solve POMDPs using locality preserving NMF can now be given:

1. **Sample belief points**: The algorithm first randomly samples a set of belief points B (set at $10,000$ in the experiments).
2. **Down-sampling:** The algorithm then sub-samples the original belief set into a smaller more manageable set B_s. To sub-sample we use the the K-nearest neighbor (KNN) algorithm and some constant δ. We include in the set B_s points that are $\geq \delta$ apart in terms of euclidean distance.
3. **Create a neighborhood graph between points**: Next, the algorithm estimates an adjacency matrix W and degree matrix D. For every belief point $b_i \in B_s$, W is computed to be:

   $$W_{i,j} = \begin{cases} 1 & \text{if } b_j \in N_k(b_i) \\ 0 & \text{otherwise} \end{cases}$$

   where $N_k(b_j)$ denotes the set of k nearest neighbors of b_j in euclidean space. The matrix W is then symmetrized as:

   $$W = (W^T + W)/2.$$

   The degree matrix is a diagonal matrix defined by the row sums of the weight matrix W:

   $$D_i = \sum_{j=1}^{|B_s|} W_{i,j}.$$

4. **Compute the graph Laplacian**: Compute the Laplacian as:

   $$L = (D - W).$$

5. **Compress the POMDP:** Given the graph Laplacian, use the LPNMF algorithm to compute $F^\dagger$. In practice, we first use the standard Euclidean NMF routine (such as the one provided in MATLAB TM) to compute some initial values for U and V. This step is useful since LPNMF is an expectation-maximization algorithm and dependent on good initial values for U and V. Then, we invoke the LPNMF algorithm $X \approx U \cdot V^T$ with input $X = B_s^T$. From the output we set $F^\dagger = U^T$ as can be derived from Equation 6. To compute F we use the pseudoinverse $F = (F^\dagger)^\dagger$. Alternatively, to approximate the orthogonality constraint similar to the orthogonal NMF algorithm in (Li et al. 2007), we approximate $I \approx F \cdot F^\dagger$ using NMF. To do this we set $X = I$, $V = (F^\dagger)^T$ and invoke the standard NMF algorithm, where we only update equation 8 and keep V constant. Finally, we compute $\tilde{R}$ and $\tilde{G}$ according to Equations 4 and 5. Enforcing orthogonality ensures that the reduced transition dynamics are non-negative and in practice gives better results than the pseudoinverse approach, which was the major motivation for the ONMF algorithm in (Li et al. 2007).
6. **Policy computation and execution**: We can compute and execute a policy as was described in the previous section on linear dimensionality reduction for POMDPs.

Experimental Results

We compared the LPNMF approach for compressing POMDPs with the standard NMF and orthogonal NMF algorithms. These methods were compared in terms of the quality of the average discounted reward $\frac{\sum \gamma^t r}{N}$ resulting from the policy as a function of the compression level (the number of dimensions). We used five standard benchmark problems from the literature: "tiger-grid", "hallway" , "hallway2", "rock-sample" and "tag-avoid". For every problem we randomly sampled $10,000$ belief points, which were subsampled to 349, 149, 506, 829 and 921 points respectively, using δ values of: $0.3, 0.5, 0.3, 0.06$ and 0.25. For all of our experiments we ran the PBVI algorithm for a fixed amount of time and measured the quality of the policy for different dimensions. We used the Perseus algorithm as the underlying PBVI algorithm (Spaan and Vlassis 2005). The λ parameter for LPNMF was chosen empirically. We observed for high dimension a small λ was required, which increased for middle dimension and then decreased again for very low dimensions. Our results are summarized in Figures 3, 4, 5 , 6, and 7. Some of the empirical λ values chosen for one of the problems are shown in Table 1. In the tiger-grid, hallway, rock-sample, and tag-avoid problems, the locality-preserving NMF algorithm clearly outperformed the other two methods, particularly at higher levels of compression. The differences are less significant in the hallway2 problem, although once again, the LPNMF method is superior at higher compression levels.

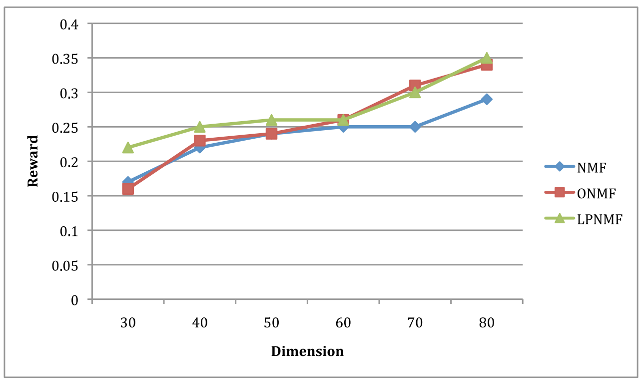

Figure 3: The hallway2 problem has $|S| = 93$, $|A| = 5$ and $|Z| = 17$. The maximum time we ran each algorithm was 60 sec. In this experiment the LPNMF approach is slightly better at lower dimensions than the regular NMF method. The state of the art ONMF approach performs worse than the NMF method.

Table 1: Empirical λs for the hallway2 domain.

k	30	40	50	60	70	80
λ	2	2	5	1.5	0.3	0.2

To verify that our compression can compute better policies in less time, we looked at the "tag-avoid" benchmark POMDP problem. The tag-avoid problem is an order of magnitude larger than the other four POMDPs and better demonstrates the realistic benefits of such compressions. Table 2 shows some of the results, where we reduced the dimension of the tag-avoid problem from 870 to 150 states and still got a reasonable solution, which is half-way between the optimal and what the Perseus algorithm achieves for the same amount of time. [1]

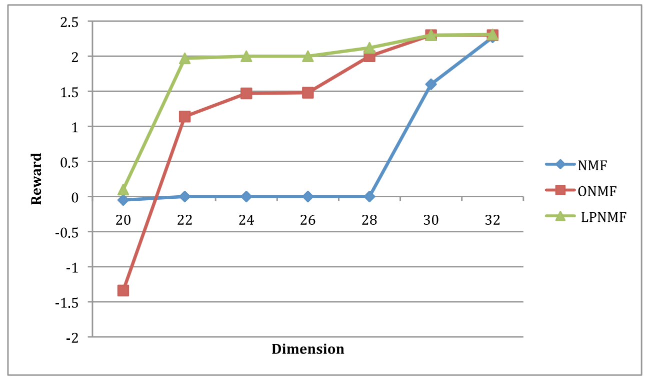

Figure 4: The tiger-grid POMDP has $|S| = 36$, $|A| = 5$ and $|Z| = 17$. The maximum time we ran each algorithm was 70 sec. The LPNMF approach significantly outperforms the plain NMF and ONMF approaches.

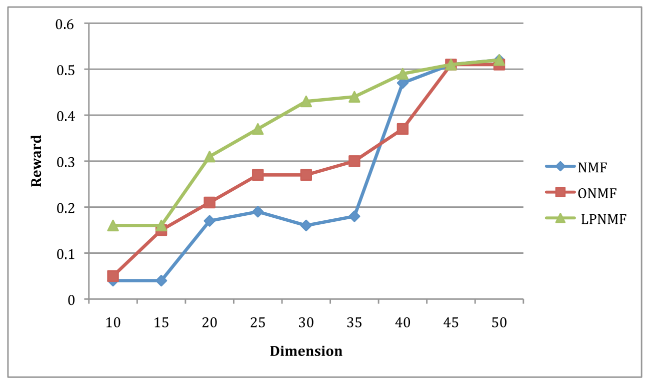

Figure 5: The hallway problem has $|S| = 60$, $|A| = 5$ and $|Z| = 21$. The maximum time we ran each algorithm was 30 sec. The LPNMF approach again allows for higher compression than plain NMF and ONMF.

Conclusions and Future Work

In this paper, we proposed a novel framework for dimensionality reduction of POMDPs based on locality preserving non-negative matrix factorization. The main advantage of the proposed approach is that it constructs a linear compression of the belief space that preserves the local geometry of the belief simplex. The experimental results show significantly improved performance compared to the state of the art orthogonal non-negative matrix factorization on sample benchmark POMDPs. This research can be extended in many ways. One way to improve the scalability of the proposed approach to larger POMDPs is to combine matrix factorization methods with low-rank matrix approximation methods, such as Kronecker decomposition. An analytical characterization of the loss in solution quality using matrix factorization methods is desirable. Finally, other nonlinear

[1] All of our experiments were done on a 2.6 GHz Core 2 Duo Intel Mac.

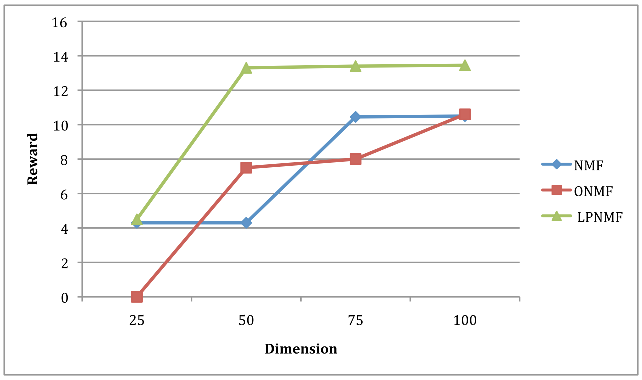

Figure 6: The rock-sample problem has $|S| = 257$, $|A| = 9$ and $|Z| = 2$. The maximum time we ran each algorithm was 50 sec. The graphs show the results for different dimensions and compression algorithms. The LPNMF approach allows for deeper compression than both the NMF and ONMF algorithms.

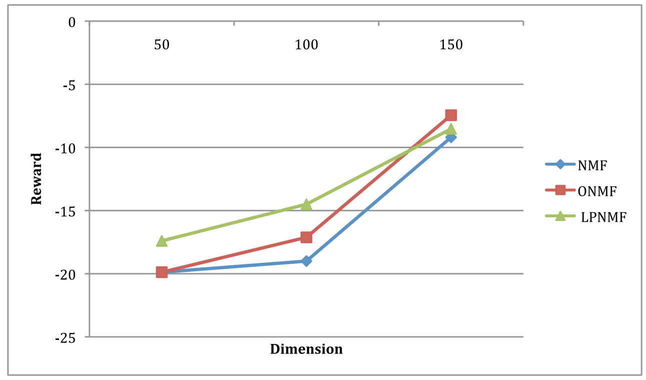

Figure 7: The tag-avoid problem has $|S| = 870$, $|A| = 5$ and $|17| = 2$. The maximum time we ran each algorithm was 2200 sec. The graphs show the results for different dimensions and compression algorithms. The LPNMF approach allows for deeper compression than both the NMF and LPNMF algorithms.

Table 2: PBVI versus LPNMF.

	Time (sec)	$\frac{\sum \gamma^t r}{N}$
tag-avoid $|S| = 870$, $|A| = 5$, $|Z| = 17$		
Perseus-PBVI	2300 500	-6.5 -13.0
LPNMF ($|\tilde{S}| = 150$)	500	-9.0

dimensionality reduction methods need to be investigated as well.

Acknowledgments

This research was supported in part by the National Science Foundation under grants NSF IIS-0534999 and NSF IIS-0803288.

References

Belkin, M., and Niyogi, P. 2003. Laplacian Eigenmaps for dimensionality reduction and data representation. *Neural Computation* 6(15):1373–1396.

Cai, D.; Xiaofei He, X. W.; Bao, H.; and Han, J. 2009. Locality preserving nonnegative matrix factorization. In *IJCAI*.

Cassandra, A. R.; Littman, M. L.; and Zhang, N. L. 1997. Incremental pruning: A simple, fast, exact method for partially observable Markov decision processes. In *Uncertainty in Artificial Intelligence (UAI*.

Cassandra, A. 1998. *Exact and Approximate Algorithms for Parttially Observable Markov Decision Processes*. Ph.D. Dissertation, Brown Univeristy.

Chung, F. R. K. 1997. *Spectral Graph Theory (CBMS Regional Conference Series in Mathematics, No. 92) (Cbms Regional Conference Series in Mathematics)*. American Mathematical Society.

Hsu, D.; Lee, W.; and Rong, N. 2007. What makes some POMDP problems easy to approximate. In *NIPS*.

Kaelbling, L. P.; Littman, M. L.; and Cassandra, A. R. 1998. Planning and acting in partially observable stochastic domains. *Artificial Intelligence* 101:99–134.

Lee, D. D., and Seung, H. S. 1999. Learning the parts of objects by non-negative matrix factorization. *Nature* 401(6755):788–791.

Lee, D. D., and Seung, H. S. 2001. Algorithms for non-negative matrix factorization. In *NIPS*, 556–562.

Li, X.; Cheung, W. K. W.; Liu, J.; and Wu, Z. 2007. A novel orthogonal nmf-based belief compression for POMDPs. In *ICML '07: Proceedings of the 24th international conference on Machine learning*, 537–544. New York, NY, USA: ACM.

Lusena, C.; Goldsmith, J.; and Mundhenk, M. 2001. Non-approximabilty results for Partially Observable Markov Decision Processes. *J. Artif. Intell. Res. (JAIR)* 14:83–103.

Pineau, J.; Gordon, G. J.; and Thrun, S. 2006. Anytime point-based approximations for large POMDPs. *Journal of Artificial Intelligence Research* 27:335–380.

Poupart, P., and Boutilier, C. 2002. Value-directed compression of POMDPs. In *In NIPS 15*, 1547–1554. MIT Press.

Smallwood, R. D., and Sondik, E. J. 1973. The optimal control of Partially Observable Markov Processes over a finite horizon. *Operations Research* 21(5):1071–1088.

Smallwood, R. D., and Sondik, E. J. 1978. The optimal control of Partially Observable Markov Processes over the finite horizon: Discounted costs. *Operations Research* 26(2):282–304.

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