Exemplar Retrieval for Enhancing RNN-based POMDP Models

Jack Lindsey
jacklindsey@stanford.edu
CS 238 Final Project

Abstract

Solving Partially Observed Markov Decision Processes (POMDPs) requires aggregation of multiple incomplete state observations through time. Recurrent neural networks (RNNs) offer a way to learn to perform this aggregation effectively when computing or approximating Bayes-optimal belief state updates is intractable. However, RNNs trained through gradient-based methods are known to have limited ability to learn to effectively aggregate information over many time steps, as is necessary in POMDPs with highly noisy or incomplete observations. We propose a model that mitigates this issue by augmenting the RNN embedding vector with its nearest neighbors in a stored finite bank of exemplar vectors. This has the effect of discretizing the possible embedding values along the newly added dimensions in the RNN embedding space. The exemplar vectors are learned along with the rest of the model, allowing for optimization of this partial discretization protocol. We find, through experimentation on the Mountain Hike benchmark task with varying levels of observation noise, that this approach aids model performance as observation noise grows. These results suggest a new approach to enhancing the performance of RNN-based POMDP models in tasks where observations are severely limited.

1 Introduction and Related Work

Deep reinforcement learning techniques have exhibited impressive performance on a variety of tasks, but their application to POMDPs is less well-studied. One popular approach, the Deep Recurrent Q-network (DRQN) of [1] uses a recurrent neural network (RNN) to aggregate observations from earlier time points. Such information aggregation is necessary for POMDPs, in which the optimal action to take is not able to be inferred from the current state observation alone. This method was refined further in the action-specific deep recurrent Q-network model of [2]. Though these methods consider Q-learning, the concept of maintaining an RNN-based embedding of the state/action trajectory can in principle be used with any deep reinforcement learning algorithm to apply it to POMDPs.

In [3], Igl et al. improve the RNN-based approach to POMDPs (in their case, they consider an RNN-based actor-critic model) by imposing additional structure on RNN updates. Their RNN is incentivized to explicitly represent inference of a belief distribution over a partially observed state space, whose transition dynamics are learned using a generative model with an auxiliary loss. They introduce the "Mountain Hike" task, a continuous control task in which agents are meant to navigate along a particular path (dubbed a mountain ridge) but receive noisy observation of their current location. The difficulty in this task is estimating the current position from the noisy observations. They find that their method outperforms an RNN-based baseline without the generative model structure.

All these methods confront the computational challenge of recurrently computing uncertainty in the current estimation of state. Igl et al. use a particle filtering approach, while DRQN represents this belief state uncertainty implicitly. As the space of possible probability distributions over the state is very high-dimensional, it can be difficult to learn an appropriate representation given limited training data.

Our model also draws techniques from the recent work of Ritter et al. in [4]. Their model incorporates an episodic memory module into a recurrent neural network-based meta-learning system by allowing the agent to reinstate previously experienced RNN hidden states, which are stored in a memory but accessed in differentiable fashion and updated as the model is trained end-to-end. Reinstatement of stereotyped RNN hidden activations allows the model to learn new policies much faster, as the retrieval allows the model to efficiently leverage previously learned structure even when encountering entirely novel states. In this model, memory retrieval is based on contextual cues that indicate which task the agent is currently facing, and the memories correspond to previously experienced RNN activation patterns. In particular, the model finds the activation patterns corresponding to the k nearest neighbors in the memory (according to the contextual cue index) to the current context. By repeatedly reinstating samples from a constrained of activity patterns through training, the model has much more training data with which to learn how to use these particular activity patterns effectively. Our model aims to leverage a similar effect.

2 Motivation and Model Outline

This work proposes a new way to maintain an implicit estimate of the true state in a POMDP given noisy state observations. We are motivated by the fact that the DRQN model and the more sophisticated model of Igl et al. rely heavily on gradient updates to a recurrent neural network in order to learn the appropriate representation of uncertainty. These networks must not only learn an effective encoding of high-dimensional states but must also be able to learn to represent sequences of states, which grows exponentially more difficult as the length of the sequence that must be encoded in order to aggregate sufficient state information increases. Recent work has shown that recurrent neural networks in practice have considerable difficulty in learning representations of arbitrary-length sequential information. In [5], Bai et al. note that ”the ’infinite memory’ advantage of RNNs is largely absent in practice,” and Miller and Moritz prove in in [6] that in many cases a recurrent neural network trained via gradient descent can be well approximated by a feedforward neural network. This phenomenon is, in essence, a manifestation of the “vanishing gradient” effect which makes it difficult for long-term dependencies to be learned effectively via backpropagation through time. As Miller and Moritz note, even though more sophisticated recurrent units like LSTMs and GRUs were developed to address this problem, they do not do so completely. Thus, we might expect that approaches to solving POMDPs that rely on fully differentiable recurrent neural network models will, as observation noise increases, have increasing difficulty capturing the long-term dependencies necessary to maintain an accurate representation of the belief state.

We propose a mechanism which aims to mitigate this suboptimality of RNN-based POMDP solvers in when the state observation is highly noisy or incomplete. In particular, we partially discretize the space of RNN embeddings by partitioning the state space into disjoint regions and augmenting the state embedding by concatenating it with additional embedding vectors associated with its region. Thus, the mapping from a sequence of observations to a state embedding varies both continuously (within each region) and discretely (across regions, since the augmentation vector changes discontinuously at regional boundaries). The discrete component allows the network to effectively learn how to act based on coarse estimates of state uncertainty, as there are few enough regions that each is visited reasonably often as the agent acts in the environment. The continuous component of the representation allows for some flexibility to fine-tune the RNN representation beyond what is allowed by the discrete set. The discretization is implemented by maintaining a finite set of exemplar points in the state space, which partition the state space into disjoint regions – each region is defined by the identity of its k nearest neighbors in the set. The positions of these exemplars are learned along with the rest of the model, allowing them to be optimized in a productive way. One expects that the model will learn to place these points at canonical locations in state space, motivating the use of the word “exemplar.”

Our model’s activation retrieval mechanism differs from that of Ritter et al. in several important ways. First, it retrieves exemplars, which may or may not lie near previously experienced states, rather than memories. Second the exemplar retrieval is indexed by the current state embedding rather than an auxiliary contextual cue (since no additional cue is available in the general POMDP setting). We predict that despite these significant differences from the model of Ritter et al., our model’s repeated reinstatement of stereotyped patterns of activity will still guide it to learn how to use these particular activity patterns effectively. We hypothesize that repeated exposure to these exemplar patterns will tend to strengthen the gradient signal that passes through them during backpropagation, mitigating the vanishing gradient problem and enabling the model to learn to recurrently updated representations of state uncertainty more effectively. Hence, we expect that the benefits of the model, if observable, will manifest most clearly for POMDPS with a high degree of observation noise.

3 Methods

All of our experiments are performed on the Mountain Hike task with varying levels of observation noise, $\sigma \in \{0.01, 0.05, 0.25, 1.0\}$. In the task, an agent is meant to navigate along an elevated "mountain ridge" while receiving noise-corrupted observations of its current state at each time step. The state space consists of 2D coordinates (x, y) in a bounded box, and actions consist of intended movements $(\Delta x, \Delta y)$ where $||(\Delta x, \Delta y)|| \leq 0.5$. Transitions are stochastic with mean corresponding to the intended action and standard deviation equal to 0.25. State observations are corrupted with Gaussian noise with standard deviation σ. Rewards are given as $R_t = f(x_t, y_t) - 0.01 \cdot ||a_t||$ where f represents the elevation of a given point in the environment.

As Mountain Hike is a continuous control task, Q-Learning methods relying on a discrete action space cannot be applied. Hence, all our models are based on a recurrent variant of the Advantage Actor Critic model (A2C) from [7], which is well suited to problems with continuous action spaces. A2C uses one neural network (the "actor") to perform policy gradient updates which are based not on empirically observed rewards but rather based on estimated expected rewards of state-action pairs as computed by another trained neural network (the "critic"). The critic is trained such that its value estimates match empirically observed Q-values of state-action pairs. The overall architecture of this baseline model is similar to that of the Deep Recurrent Q Network model, but with the A2C algorithm replacing Q-learning, and without any convolutional layers needed at the start of the model in our case (since the Mountain Hike task has low-dimensional state space). More concretely, the model feeds the given observation and the previous action through two linear feedforward neural network layers of width 64, each with ReLU activations. The result is then fed to a layer of generalized recurrent units (GRU) (introduced in [8]) of width N. The GRU layer also receives its state at the previous time step as input, allowing for recurrent updates. The GRU layer outputs an embedding to an M-dimensional state space (we used $M = N = 128 \cdot 6 = 768$ in all experiments). In the "RNN-A2C" baseline model, this embedding vector is fed in as the input to the A2C algorithm. Code for defining the task environment and implementing the RNN-A2C baseline was adapted from Igl et al.'s implementation of their generative model approach at https://github.com/maximilianigl/DVRL. To quantify the importance of recurrence at different levels of observation noise, we also experimented with a pure A2C baseline using no recurrence, in which the mapping from observation to embedding consisted solely of the two initial feedforward layers.

Our exemplar-based approach (which we will refer to as RNN-Exemplar-A2C) differs from the above in the following manner. 50 random points are initialized in the exemplar bank by projecting observations into embedding space via the model's randomly initialized weights for 50 time steps with the model following random actions. Following this initialization phase, each time the model recurrently embeds the current observation and GRU context into a state embedding (or size 128), it concatenates this embedding with the embeddings of the $k = 5$ nearest exemplars in the exemplar bank (for a total embedding size of $128 + 5 \cdot 128 = 768$. The GRU then takes in this augmented vector as input and computation proceeds as in the baseline RNN-A2C model above. The model is updated end-to-end via backpropagation, and the locations of the exemplars, once initialized are considered as learnable parameters that are updated along with the rest of the model's weights. All optimization was performed using RMSProp with $\alpha = 0.99$. All models were trained for 2.5 million iterations. Note that the architecture matches that of the RNN-A2C model to admit fair comparison – both models have comparable numbers of parameters (the RNN-Exemplar-A2C model has slightly fewer due to the nonparameteric augmentation computation). See Figure 1 for a visualization of the model architecture.

4 Results

We test the A2C, RNN-A2C, and RNN-Exemplar-A2C models on the Mountain Hike task on different settings of the noise parameter: $\sigma = 0.01, 0.05, 0.25$, and 1.0. See Figure 2. We observe a few trends. First, as would be expected, the feedforward A2C model performs competitively with the other models for low noise values. This is reasonable, as the Mountain Hike task becomes a fully

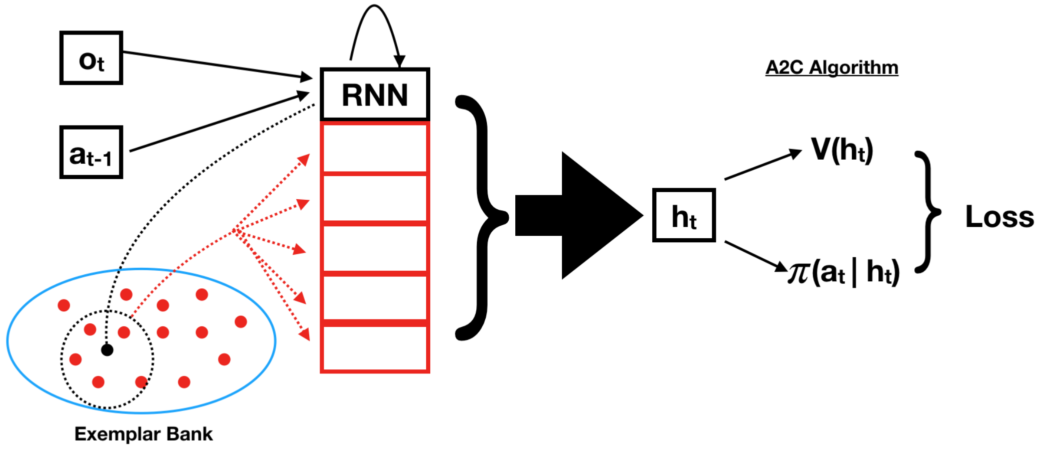

Figure 1: A schematic depicting the architecture of the RNN-Exemplar-A2C model. First, the noisy state observation and previous action are mapped to an embedding space. The k nearest neighbors to this embedding in the exemplar bank are retrieved and concatenated onto the embedding vector. The augmented vector is used as the input to a GRU-based recurrent neural network, which also receives as input its own output from the last time step. The RNN outputs a vector h_t which is fed in as input to the advantage actor-critic (A2C) model, in which an "actor" neural network learns a policy through policy gradient updates based on the reward estimated by the "critic network," which is trained to accurately estimate the value of given values of h_t. The RNN-A2C model is the same except that it lacks the exemplar-based augmentation step. The A2C model is the same except that its lacks the RNN, feeding state/action embeddings directly into the actor and critic networks. Format based on a figure in [3].

observed MDP when $\sigma = 0$, in which case the optimal policy should in theory be computable via feedforward computation from the current state observation (of course, differences in architecture and number of parameters may still cause A2C and RNN-A2C to give different results in the zero-noise case). As σ increases, the A2C model becomes unable to learn the task and exhibits unstable learning trajectories. This is expected, as the error in state estimation by a feedforward model which only accesses one noisy state observation will grow large as the noise increases.

Second, the RNN-Exemplar-A2C model grows increasingly competitive as σ grows, eventually overtaking all other models in performance. For $\sigma = 0.01$ it is asymptotically outperformed by RNN-A2C. For $\sigma = 0.05$ it matches RNN-A2C but learns slower. For $\sigma = 0.25$ the same is true but the learning speed disparity is smaller. Finally, for $\sigma = 1.0$, RNN-Exemplar-A2C significantly outperforms the other models.

An additional interesting observation is that a modified version of the RNN-Exemplar-A2C model, in which the RNN state was not *augmented* by the retrieved exemplar vectors but instead *replaced* by them, could not achieve task performance better than that achieved with random weights, for any value of noise. This result suggests that while the partial discretization of the embedding space achieved by the RNN-Exemplar-A2C model can be helpful in the presence of high noise, complete discretization is not a viable solution, at least given the small exemplar bank size (and correspondingly coarse embedding space discretization) used in our experiments.

5 Discussion

Our experiments suggest that augmenting the activations in a recurrent neural network with its the nearest neighbors retrieved from a bank of exemplar patterns can allow it to learn better policies in POMDPs with high observation noise. In the low observation noise regime, this augmentation does not appear to help performance and even hurts it slightly (we presume the slight reduction in performance arises from the fact that in the low noise regime, the dimensions in embedding space spent on the augmentation are better used as part of a higher-dimensional RNN representation).

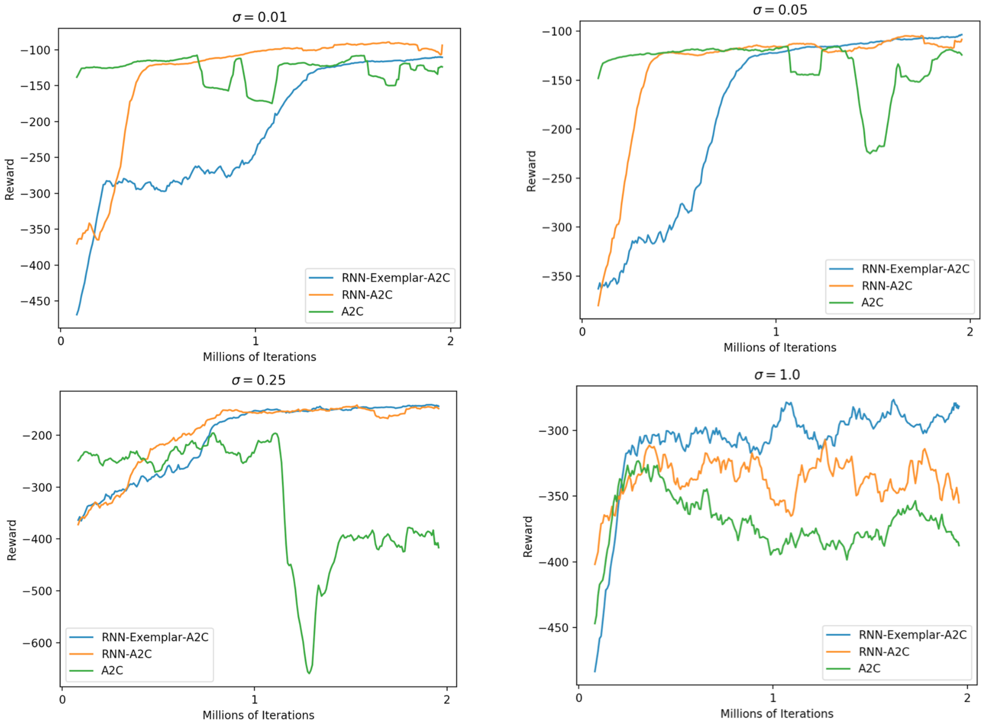

Figure 2: Performance of the A2C, RNN-A2C, and RNN-Exemplar-A2C models on Mountain Hike for different values of σ. The relative performance of RNN-Exemplar-A2C compared to the other models improves for higher noise values, while the feedforward A2C model becomes less competitive as σ increases.

This result suggests a umber of follow-up research directions. First, it should be checked whether the performance advantage of RNN-Exemplar-A2C in high-noise regimes does in fact result from the hypothesized phenomenon of mitigating the vanishing gradient problem. This could be checked empirically by tracking the magnitude and directional consistency of gradient updates as training progresses, in the RNN-Exemplar-A2C and RNN-A2C models. Second, the effect of various hyperparameters in the RNN-Exemplar-A2C model should be characterized. Time and resource constraints prevented thorough optimization of and experimentation with the choice of the size of the exemplar bank. Increasing this size significantly, which likely requires improving the efficiency of our implementation of the nearest neighbors retrieval, could allow for much denser coverage of embedding space by the exemplar bank, which might improve performance significantly. The choice of the number k of exemplars retrieved at each iteration is also likely to have significant impact on performance.

A number of reasonable extensions to the model are also worth considering. For instance, each of the k exemplars retrieved was weighted equally in our model. One could potentially also augment the embedding space with information about the proximity of the exemplars to the original state embedding, allowing the model to weight the top nearest neighbors more heavily. One could also imagine replacing infrequently accessed exemplars with new ones during training, perhaps initializing the replacement exemplars to correspond to the RNN embeddings of recent observation/action pairs (which would bring the model closer to the memory retrieval paradigm of Ritter et al.). Additionally, it would be worth exploring alternative methods of augmenting the embedding space with information about the exemplars besides simple concatenation. Finally, the performance of the exemplar approach should be assessed on a wide variety of tasks and in combination with a wide variety of base RL algorithms (i.e. others besides A2C).

Overall, this work suggests a promising new approach to enhancing the performance of RNN-based POMDP models when observations are very noisy or incomplete, through partial discretization of the RNN embedding space via exemplar retrieval.

References

[1] Matthew Hausknecht and Peter Stone. Deep recurrent q-learning for partially observable mdps. 2015.

[2] Pengfei Zhu, Xin Li, Pascal Poupart, and Guanghui Miao. On improving deep reinforcement learning for pomdps. *arXiv preprint arXiv:1804.06309*, 2018.

[3] Maximilian Igl, Luisa Zintgraf, Tuan Anh Le, Frank Wood, and Shimon Whiteson. Deep variational reinforcement learning for pomdps. *arXiv preprint arXiv:1806.02426*, 2018.

[4] Samuel Ritter, Jane X Wang, Zeb Kurth-Nelson, Siddhant M Jayakumar, Charles Blundell, Razvan Pascanu, and Matthew Botvinick. Been there, done that: Meta-learning with episodic recall. *arXiv preprint arXiv:1805.09692*, 2018.

[5] Shaojie Bai, J Zico Kolter, and Vladlen Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*, 2018.

[6] John Miller and Moritz Hardt. When recurrent models don't need to be recurrent. *arXiv preprint arXiv:1805.10369*, 2018.

[7] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, pages 1928–1937, 2016.

[8] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.