Explicit Sequence Proximity Models for Hidden State Identification

Anil Kota, Sharath Chandra and Parag Khanna
Visvesvaraya National Institute of Technology, India
{anilkota1997, sharathraparthy, paragkhanna1}@gmail.com

Torbjørn S. Dahl
InstaDeep Ltd./University of Plymouth, UK
t.dahl@instadeep.com

1 Introduction

Sequence similarity is a critical concept for comparing short- and long-term memory in order to identify hidden states in partially observable Markov decision processes. While connectionist algorithms can learn a range of *ad hoc* proximity functions, they do not reveal insights and generic principles that could improve overall algorithm efficiency.

Our work uses the instance-based Nearest Sequence Memory (NSM) [5] algorithm as a basis for exploring different explicit sequence proximity models including the original NSM proximity model and two new models, temporally discounted proximity and Laplacian proximity. The models were compared using three benchmark problems, two discrete grid world problems and one continuous space navigation problem. The results show that more forgiving proximity models perform better than stricter models and that the difference between the models is more pronounced in the continuous navigation problem than in the discrete grid world problems.

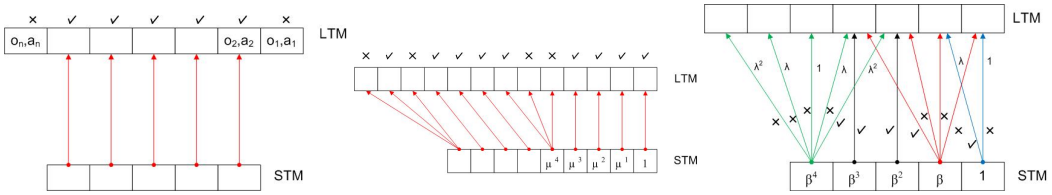
The hidden state identification problem can to some extent be avoided by concatenating information from several time steps in the learning input [6, 8] so that temporal features can be extracted directly without the use of memory. Though successful in some domains, this approach is limited by the quick growth in the size of the input space. Hidden state estimation fundamentally involves comparing recent observations and actions from the current episode, stored in STM, to observations and actions from earlier episodes stored in LTM. It is possible to estimate the current state by maintaining an explicit probability distribution across states, a *belief state* [2]. The solutions following this approach, however, have required knowledge of a discrete state space and state transition function which is unrealistic in many applications. Recurrent neural networks (RNNs) are currently the most successful way of encoding sequence information and using it for hidden state identification and reward optimization [1, 3] though temporal convolutional networks now offer a potentially more powerful, hierarchical, non-recurrent, alternative [4]. The HSOME algorithm [7] uses temporal abstraction, hierarchy and a generic sequence proximity function to encode temporal features in a Kohonen self-organizing map and use these for reward optimisation. This encoding avoids the biologically infeasible back-propagation mechanism but increased learning speed has not been demonstrated. Though the HSOME algorithms could be used for this work, we have for simplicity, focused on an instance-based approach.

2 Sequence Proximity Models

NSM is based on the *K-nearest neighbor* algorithm. During an episode, the agent records the actions, observations and reward values at each step in STM. During action selection, it compares STM with previous episodes stored in LTM, and identifies K matches in LTM that most closely resemble its

32nd Conference on Neural Information Processing Systems (NIPS 2018), Montréal, Canada.

current situation. From these K instances, the agent then calculates and selects the action with the highest mean discounted reward. The NSM model of proximity is the length of the unbroken chain backwards in time, of steps that are identical in STM and LTM. This is a fairly strict proximity model as a single point of deviation will terminate the chain of matches. Any matching nodes further back in time are not taken into consideration. We evaluated the relative performance of the NSM proximity model by comparing it to two less strict models, the discounted proximity model and the Laplacian proximity model. The three models are presented graphically in Figure 1.



(a) NSM proximity (b) Discounted proximity (c) Laplacian proximity

Figure 1: The three proximity models used in this work. Ticks indicate matches while crosses indicate mismatches. Figure 1(a), *NSM proximity*, counts subsequent matches between STM and LTM. Figure 1(b), *discounted proximity*, uses a sum of discounted values ignoring non-matching values in LTM. Figure 1(c), *Laplacian proximity*, uses a sum of values from a sequence of Laplacian distributions across LTM centered on each node.

The discounted model introduces a temporal discount factor, μ, for each step backwards in time away from the current STM step. It also skips mismatching LTM steps and continues to evaluate the next step instead of terminating the sequence on the first mismatch. This model emphasizes early matches and can pick up on similarities beyond the first mismatch. The Laplacian proximity model combines a global discount factor, β, with a local discount factor, λ, which is used to discount the contribution of a match in accordance with its distance from the corresponding step in STM. In this model the contribution of the match is discounted by the factor μ for each STM step away from the current step. This produces the globally discounted value which is further discounted by the factor λ for each LTM step away from the corresponding STM step the match is found. Each LTM step are only allowed to be matched once. This model provides further flexibility in the matching by looking both forward and backwards in time for a possible match. It also punishes gaps unlike the discounted model.

3 Experiments and Results

All three proximity models were evaluated on three benchmark reinforcement learning problems: the Tiegr problem, Sutton's grid world and a simulated E-puck navigation problem. The robot navigation problem was discretized using a representation with three actions: turn-left, turn-right and move forward and eight observations representing the readings of four infrared sensors. The problems are illustrated in Appendix A.

For all the experiments, the key parameters of the learning algorithms were kept the same. For the fundamental nearest sequence memory algorithm, we used $k = 8$, $\gamma = 0.9$ and $\epsilon = 0.2$. The number of subsequent observations kept in short-term memory were 20 and the number of episodes stored in long-term memory were also 20. For the discount model of proximity we also used a discount factor $\mu = 0.95$ and for the Laplacian proximity model we used a global discount factor $\beta = 0.95$ and a local discount factor $\lambda = 0.95$. For all the problems we ran 20 trials of 100 episodes with the above experimental parameters.

The results of our experiments are presented as plots in Appendix B. They show that less strict proximity models, such as the Laplacian model in general perform better than the stricter ones, such original NSM model. The result is particularly clear in the robot navigation problem, there is also a clear difference in the mean proximity values.

References

[1] Bram Bakker. Reinforcement learning with long short-term memory. In *Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic (NIPS'01)*, pages 1475–1482, December 2001.

[2] Anthony Cassandra, Michael L. Littman, and Nevin L. Zhang. Incremental pruning: a simple, fast, exact method for partially observable markov decision processes. In *Proceedings of the Conference on Uncertainty in artificial intelligence (UAI'97)*, pages 54–61, Providence, Rhode Island, August 1-3 1997.

[3] Matthew Hausknecht and Peter Stone. Deep recurrent Q-learning for partially observable MDPs. In *AAAI Fall Symposium on Sequential Decision Making for Intelligent Agents (AAAI-15)*, November 2015.

[4] Colin Lea, René Vidal, Austin Reiter, and Gregory D. Hager. Temporal convolutional networks: A unified approach to action segmentation. *CoRR*, abs/1608.08242, 2016.

[5] Andrew McCallum. Instance-based state identification for reinforcement learning. In Gerald Tesauro, David S. Touretzky, and Todd K. Leen, editors, *Proceedings of the 1994 Conference on Advances in Neural Information Processing Systems (NIPS 7)*, pages 377–384, 1994.

[6] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Venes, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, Charles Beattie Stig Petersen, Amir Sadik, Ioannis Antonoglo, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

[7] Georgios Pierris and Torbjørn S. Dahl. Learning robot control using a hierarchical SOM-based encoding. *IEEE Transactions on Cognitive and Developmental Systems*, 9(1):30–43, 2017.

[8] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Bake, Matthew Lai, Adrian Bolton, Yutian Chen, Timothy Lillicrap, Fan Hui, Laurent Sifre, George van den Driessche, Thore Graepel, and Demis Hassabis. Mastering the game of go without human knowledge. *Nature*, 550:354–359, 2017.

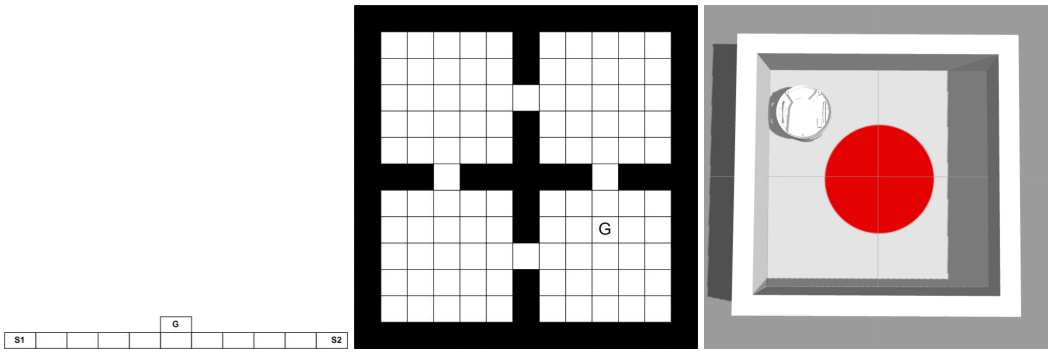
[9] Richard Sutton, Doina Precup, and Satinder Singh. Between mdps ans semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence*, 112:181–211, 1999.

A Problems

The three problems used for this work were two grid worlds and one mobile robot navigation task in a continuous domain. The grid worlds were a T-maze and a world with four rooms initially presented by Sutton *et al.* [9]. The worlds are presented in Figure 2.

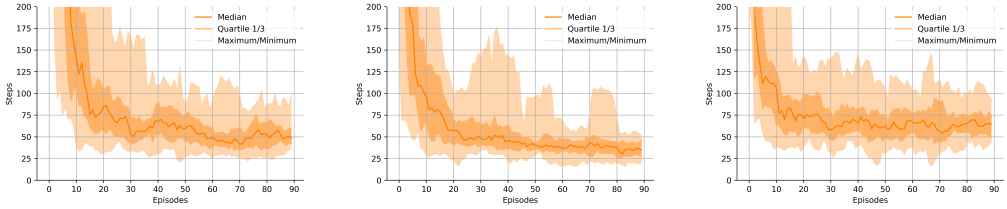
B Performance

Below are plots of the performance and proximity values for the different benchmark problems and algorithms discussed in the paper are presented below as well as a table summarizing the Mann-Whitney U-test results for comparing the performance of the different proximity models. The proximity values are the sum of all the proximity values in the K-nearest sequences for each step.



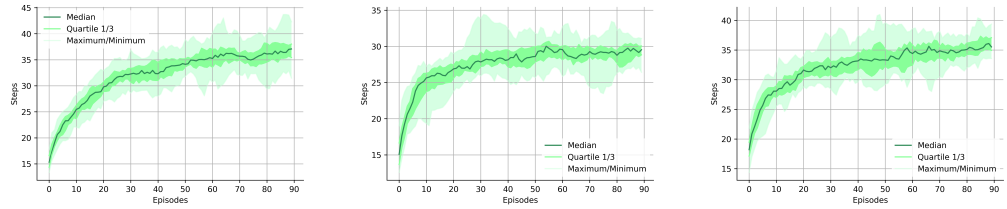
(a) T-maze (b) Sutton's world (c) Robot navigation

Figure 2: The three proximity models used in this work. Ticks indicate matches while crosses indicate mismatches. Figure 2(a) shows the *T-maze*, a very simple POMDP. Figure 2(b) shows *Sutton's grid world*, a POMDP with a large number of hidden states. Figure 2(c) shows the *robot navigation* problem, a continuous POMDP



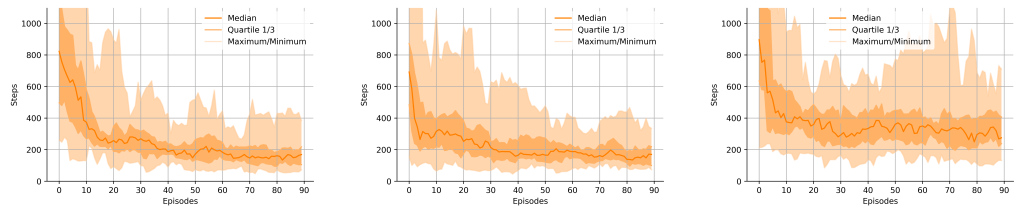
(a) McCallum proximity (b) Discounted proximity (c) Laplacian proximity

Figure 3: The performance of the different proximity models on the Tiger problem



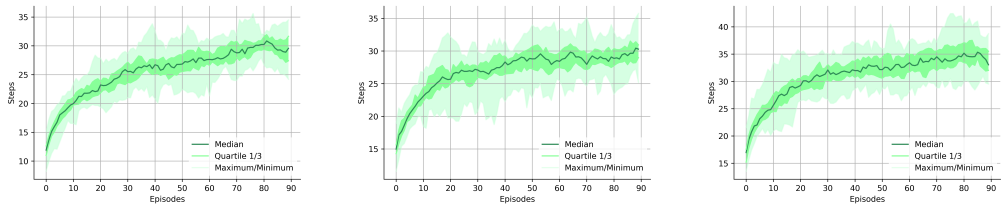
(a) McCallum proximity (b) Discounted proximity (c) Laplacian proximity

Figure 4: The proximity values produced on the Tiger problem



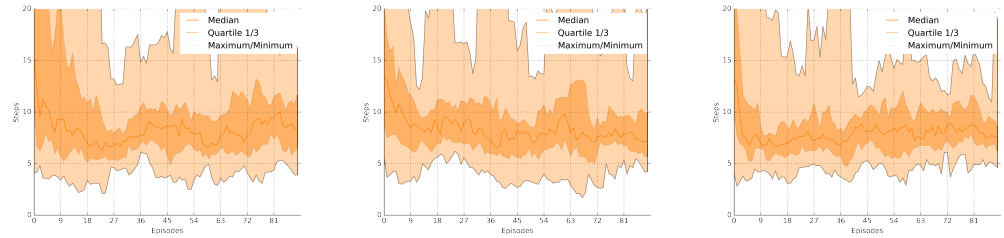
(a) McCallum proximity (b) Discounted proximity (c) Laplacian proximity

Figure 5: The performance of the different proximity models on Sutton's grid-world problem



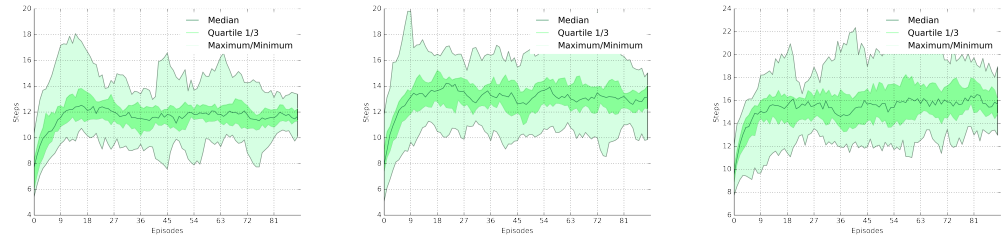
(a) McCallum proximity (b) Discounted proximity (c) Laplacian proximity

Figure 6: The proximity values produced on the Sutton's grid-world problem



(a) McCallum proximity (b) Discounted proximity (c) Laplacian proximity

Figure 7: The performance of the different proximity models on the E-puck navigation problem



(a) McCallum proximity (b) Discounted proximity (c) Laplacian proximity

Figure 8: The proximity values produced on the E-puck navigation problem

Problem	Comparison	1	10	20	50	80	100
Tiger	Discounted < McCallum	0.241	0.208	0.347	0.357	0.023	0.950
	Laplacian < McCallum	0.886	0.780	0.107	0.462	0.511	0.955
	Laplacian < Discounted	0.857	0.965	0.672	0.857	0.973	0.984
Sutton	Discounted < McCallum	0.290	**0.026**	0.672	0.733	0.373	0.596
	Laplacian < McCallum	0.484	0.965	0.995	0.988	0.977	0.997
	Laplacian < Discounted	0.691	0.993	0.999	0.996	0.999	0.983
E-Puck	Discounted < McCallum	0.961	0.307	0.332	0.643	0.419	0.633
	Laplacian < McCallum	0.894	**0.034**	0.114	0.707	0.101	0.608
	Laplacian < Discounted	0.388	0.102	0.248	0.511	0.164	0.644

Table 1: Mann-Whitney U-test results o