Making and Improving Predictions of Interest Using an MDP Model

(Extended Abstract)

Yunlong Liu[1] Yifeng Zeng[1,2] Hexing Zhu[1] Yun Tang[1]
[1]Department of Automation, Xiamen University, Xiamen, China
[2]School of Computing, Teesside University, UK
ylliu@xmu.edu.cn yifeng.zeng.dk@gmail.com hxzhu_xmu@foxmail.com tydqhqy@qq.com

ABSTRACT

In many cases, building a generative model is difficult and unnecessary since we may be only interested in making predictions of some certain situations. In this paper, we model the dynamics of predictions of interest, called *prediction profile*, through a Markov decision process (MDP) and accordingly make the predictions using the learned model. We further adapt the entropy concept to measure prediction accuracy of the learned MDP model and provide important guidelines for strategically expanding events of interest with the purpose of improving the predictions. We conduct experiments to demonstrate the performance of our techniques.

Keywords

Predictions of Interest; Markov Decision Process; Accuracy

1. INTRODUCTION

It is known that learning a *generative model* that can make predictions for *all* possible futures is often intractably complex in a partially observable, stochastic environment of high dimensions. In many cases, it is more interesting to make predictions for a small set of future events. This motivates the developments of a *non-generative* model in making a small set of predictions of interest [1, 6, 11, 12, 13, 14]. The set of predictions of interest is usually denoted as a *prediction profile* [12, 14], which is a vector of predictions for the *tests* of interest, and a *test* is a sequence of action-observation pairs to happen in the future that can be used to describe the events of our interests.

Recent work on non-generative models usually builds partially observable models by using prediction profiles as observation representations [12, 14]. Such a treatment leads to the problems of time-consuming, local minima, etc., and prior knowledge of the system is often required.

Observing that when the tests of interest contain the set of *core tests* [4], the prediction profiles are *sufficient statistics* of actions and observations in the past and can serve as states in the underlying system. Inspired by this, we present a Markov decision process (MDP) based approach for making predictions of tests of interest by using prediction profiles as state representations. The MDP approach is demonstrated to be far more efficient and accurate in making predictions. We also demonstrate that the new MDP based prediction model consistently provides superior performance even if sufficient statistics are not guaranteed.

With the benefit from the utilization of MDP models, we can evaluate and improve the prediction accuracy of the learned MDP model before the model will be further experimented or applied in practice. Given the learned MDP model, we adopt the model entropy [8] to measure the uncertainty of state transition in the learned model, and reveal a relation between the entropy and the prediction accuracy. Such a relation further provides important guidelines for strategically expanding events of interest with the purpose of improving the predictions. This study also complements our approach when the prediction profiles are not sufficient statistics of the past, where the state transition becomes stochastic.

Appears in: *Proc. of the 16th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2017), S. Das, E. Durfee, K. Larson, M. Winikoff (eds.), May 8–12, 2017, São Paulo, Brazil.*

2. MDP BASED PREDICTION MODELS

2.1 Clustering Prediction Profiles

Given the training data, we can estimate empirically the prediction profile $\phi(h) = \{p(t_1|h), p(t_2|h), \cdots, p(t_i|h)\}$ for the tests of interest $\mathcal{T}^I = \{t_1, t_2, \cdots, t_i\}$ at history h [14] and construct a set of prediction profiles, namely $P = \{p_1, \cdots, p_n\}$, where n is the number of histories in the training data.

Due to the sample error, the estimated prediction profiles will unlikely be equal at different histories, even if the true underlying prediction profiles are identical.

To decide the set of distinct prediction profiles, we evaluate the linear independence of the composed prediction profiles P by computing the matrix rank [3, 5, 7]. As the rank computation considers the average error of matrix entries[3], compared to the statistical tests [9, 14], differentiation of the prediction profiles is more accurate. The prediction profiles that are linearly independent are classified into different groups and correspond to different true underlying prediction profiles. For each group i, a representative prediction profile pp_i is selected.

2.2 Learning MDP Models

Under the condition that the tests of interest contain the set of core tests, the prediction profile is a sufficient statistic of history so that it can represent the state of the underlying systems. Simultaneously, observing that given sufficient training data, the prediction profiles that do not appear in

the training data can be thought as never or rarely encountered. With the obtained k representative prediction profiles, we learn an MDP model to describe the dynamics of prediction profiles. The MDP specification follows:

- **States** the set of states is defined to be the set of representative prediction profiles, i.e., $\mathcal{S} = \{s_1 = pp_1, s_2 = pp_2, \cdots, s_k = pp_k\}$.
- **Actions** the set of actions is defined to be the set of action-observation pairs in the original system, i.e., $\mathcal{A}_{PP} = A \times O$.
- **State-transition function** $T : \mathcal{S} \times \mathcal{A}_{PP} \times \mathcal{S} \to [0, 1]$, where $T(s_i, \langle ao \rangle, s_j)$ is the probability of ending in s_j when an agent starts in state s_i and takes action $\langle ao \rangle$.

To learn the MDP model, we first translate the original action-observation sequences into the form of ⟨action-observation⟩-profile sequences [14]. Subsequently, we compute the transition function in the transformed data and build the MDP model. Given the learned MDP model, we can make the predictions of tests of interest at any history.

2.3 Improving the Prediction

It is noted that under the conditions that the number of the prediction profiles is finite and the prediction profiles are sufficient statistics of the histories, the learned MDP is deterministic.

PROPOSITION 1. *Given the tests of interest include the set of core tests, the dynamics of prediction profiles is deterministic.*

Consequently, if the number of prediction profiles is finite, the entry of the state-transition functions in the learned MDP model will be either 0 or 1 and the learned MDP model is deterministic. Otherwise, in many cases, other than deterministic, the transition from one prediction profile to another becomes stochastic, which will reduce the prediction accuracy of tests of interest.

To improve the prediction accuracy, we first adopt the concept of model entropy [Equation 5 of reference [8]] to measure the learned MDP model's accuracy and then make a further step to improve the learned model's prediction accuracy accordingly.

The model entropy quantifies the uncertainty of state transitions in the learned MDP model. The entropy value grows when the transitions become more stochastic, which usually means a lower prediction accuracy of the learned model [8]. Hence we can use the entropy of the learned model as one quantitive measurement of the model's prediction accuracy.

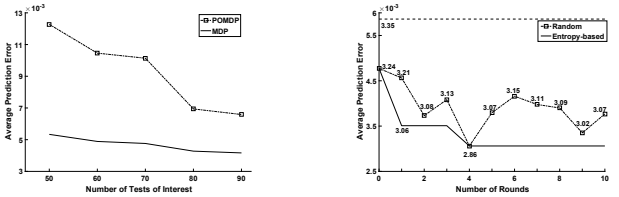
As more tests are added into the set of tests of interest, the prediction profile tends to be sufficient statistic of the history. To improve the prediction accuracy, we can expand the tests of interest in a strategic way. Intuitively we shall add the tests that can reduce the uncertainty of the transitions in the learned MDP model. Thus, the entropy of the learned MDP model can be used as guidance for choosing tests to be added into the set of tests of interest. When the entropy of the learned model is high, some tests should be added and the tests leading to a lower entropy should be chosen. We can repeat the MDP model learning and make predictions accordingly.

We design one simple iteration algorithm to properly expand the given set of tests of interest T^I. Starting with a randomly generated set of tests T^R, in each round, we iteratively sample a new test and consider using it to replace each element of T^R, then the model is relearned using $T^I \cup T^R$ as tests of interest. In each round, if the best replacement is a reduction in terms of the entropy value of the learned model, then we keep it.

3. EVALUATION

Experiments were conducted on the *PocMan* domain [2, 8, 10] and we compared the MDP based prediction technique with the POMDP approach [14]. Both approaches used the linear independence technique for clustering the prediction profiles.

We conducted two sets of experiments. The first set is to compare the two approaches' performance by varying the number of tests of interest. The learned models were evaluated in terms of prediction accuracy and model learning time. The second set is to verify the entropy based strategy for improving the accuracy of predictions of tests of interest. We compared the entropy based selection to one baseline technique that randomly replaces one element of T^R of size 10 with the sampled new test. The new T^R combined with T^I of size 50 was used to learn the corresponding model.



(*a*) (*b*)

Figure 1: Prediction errors for (*a*) different number of tests of interest; (*b*) 10 rounds in expanding tests of interest in *PocMan*.

Fig. 1 (*a*) reports the prediction accuracy of the evaluated methods when varying the number of tests of interest. The learned MDP model outperforms the POMDP approach even when the number of tests of interest is very small (less than the number of core tests). *The results support the application of the MDP model on making predictions of tests of interest even if prediction profiles are not a sufficient statistic of history.* The learning time ratio between the POMDP and MDP models reaches more than 6:1.

Fig. 1 (*b*) shows the average prediction errors of two selection methods (Random and Entropy-based) when new tests were added into the tests of interest over 10 rounds in the *PocMan* domain. The upper dashed line is the prediction error of the original MDP model. The entropy-based strategy significantly improves the prediction accuracy by adding the new tests that lead to the lowest entropy of the learned MDP model. It is observed for the random strategy, the performance is not stable over rounds and the reduction of the prediction errors is not guaranteed in most the cases. The number at each point in the figure is the (lowest) entropy at the corresponding round, which also shows that a higher entropy value results in a lower prediction accuracy.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (No. 61375077).

REFERENCES

[1] M. Dinculescu and D. Precup. Approximate predictive representations of partially observable systems. In *Proceedings of the 27th International Conference on Machine Learning (ICML)*, pages 895–902, 2010.

[2] W. Hamilton, M. M. Fard, and J. Pineau. Efficient learning and planning with compressed predictive states. *Journal of Machine Learning Research*, 15:3395–3439, 2014.

[3] M. R. James and S. Singh. Learning and discovery of predictive state representations in dynamical systems with reset. In *Proceedings of the Twenty-first International Conference on Machine Learning (ICML)*, pages 417–424, 2004.

[4] M. L. Littman, R. S. Sutton, and S. Singh. Predictive representations of state. In *Advances In Neural Information Processing Systems (NIPS)*, pages 1555–1561, 2001.

[5] Y. Liu and R. Li. Discovery and learning of models with predictive state representations for dynamical systems without reset. *Knowledge-Based Systems*, 22:557–561, 2009.

[6] Y. Liu, Y. Tang, and Y. Zeng. Predictive state representations with state space partitioning. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 1259–1266, 2015.

[7] Y. Liu, Z. Yang, and G. Ji. Solving partially observable problems with inaccurate psr models. *Information Sciences*, 283:142–152, 2014.

[8] Y. Liu, H. Zhu, Y. Zeng, and Z. Dai. Learning predictive state representations via monte-carlo tree search. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI)*, 2016.

[9] C. R. Shalizi and K. L. Shalizi. Blind construction of optimal nonlinear recursive predictors for discrete sequences. In *Proceedings of the Twentieth Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 980–987, 2004.

[10] D. Silver and J. Veness. Monte-carlo planning in large pomdps. In *In Advances in Neural Information Processing Systems 23 (NIPS)*, pages 2164–2172, 2010.

[11] E. Talvitie. Learning partially observable models using temporally abstract decision trees. In *Advances in Neural Information Processing Systems (NIPS)*, pages 827–835, 2012.

[12] E. Talvitie and S. Singh. Maintaining predictions over time without a model. In *Proceedings of International Joint Conference on Artificial Intelligence(IJCAI)*, pages 1249–1254, 2009.

[13] E. Talvitie and S. P. Singh. Simple local models for complex dynamical systems. In *Proceedings of th Twenty-Second Annual Conference on Neural Information Processing Systems (NIPS)*, pages 827–835, 2008.

[14] E. Talvitie and S. P. Singh. Learning to make predictions in partially observable environments without a generative model. *J. Artif. Intell. Res. (JAIR)*, 42:353–392, 2011.