Deep Predictive State Representation With K-Clustering

Group C
School of Data Science, City University of Hong Kong, Hong Kong, China;

Introduction

In the MDP framework, the state of the system is assumed to be completely observable by decision maker with the help of perfect observations (and hence known with certainty by the decision maker). Relaxation of this assumption yields a POMDP.
Predictive state representations (PSRs) is based entirely on predicting the conditional probability of sequences of future observations, conditioned on future sequences of actions and on the past history. Because there are no hidden states in the model, such a representation should be easier to learn from data.
In this paper, a new method is proposed. Deep Predictive State Representation With K-Clustering adds a carefully designed value function into the original model. Besides, Deep Neural Network (DNN) is applied to construct the history probe.

Preliminaries

1. POMDP

A POMDP is completely characterized by <S; A;Z; P; Q; R>, the core state space, the action space, the set of observations, the transition probability matrix, the information matrix, and the reward function.

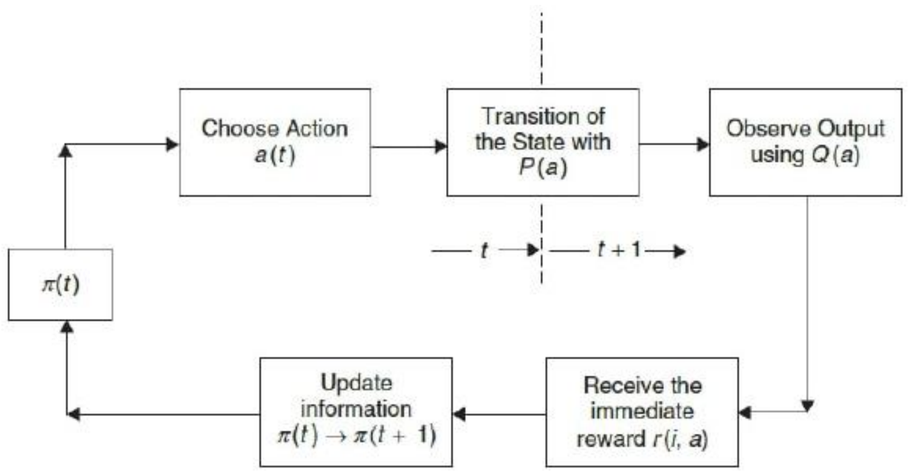

Fig. 1.

2. Predictive State Representation

We'd like to introduce some definitions before turning to the PSR.

Definition 1 *A history, denoted by h_τ, is an action-observation sequence received up to the time step τ: $h_\tau = (a_0 o_0 a_1 o_1 ... a_\tau o_\tau)$.*

Definition 2 *An action-observation sequence starting at time $\tau + 1$ is called a test, $t_\tau = (a_{\tau+1} o_{\tau+1} ... a_{\tau+k} o_{\tau+k})$*

Definition 3 *The prediction for test t given history h, $p(t|h)$, is defined as the conditional probability that $\omega(t)$ occurs, if $\sigma(t)$ is executed.*

$$p(t|h) = P((\omega(t)|h, \sigma(t))$$

Methods

Algorithm 1

Input: A set of data $D = (h, t)$, $\forall h \in H$, $\forall t \in T$, k_0, number of iteration n.
Output: g, history cluster A, probed predictions $p_f(T|H)$
Using K-means, initialize H_k, k_{labels}
for i in range(n) **do**
  for minibatch (H,T) in D **do**
    Calculate representation of H

$$h_g = g \cdot H$$

    Calculate new centers

$$C_k = \frac{\sum_{t=0}^{|h_g^k|} h_g^{k,t}}{|h_g^k|}$$

    Calculate intra-class distance

$$d_1 = \frac{\sum_{k=0}^{k0} \sum_{t=1}^{|h_g^k|} (h_g^{k,t} - C_k)^2}{k_0 \cdot |h_g^k|}$$

    Calculate distance between classes

$$d_2 = \frac{\sum_{t=0}^{k0} \sum_{j=t+1}^{k0} (C_t - C_{t+1})}{\frac{(1+k_0)k_0}{2}}$$

    Calculate distance of h_g^k with the same t

$$d_3 = \sum_{t=1}^{|T|} (h_g^1 - h_g^2)^2,$$

    h_g^1,h_g^2 are randomly chosen with the same t.
    Calculate the loss function

$$V = d_1 - d_2 + d_3$$

    Update g to minimize V.
  end for
end for

Experimental Results

1. Experiments Environment

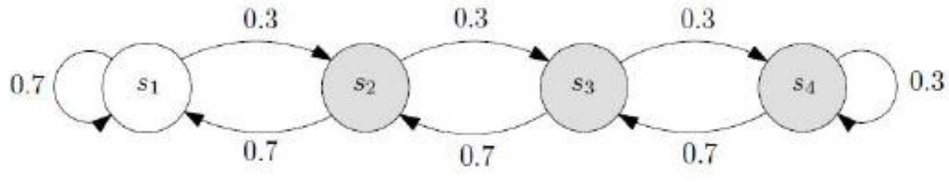

Fig. 2. Tunnel World

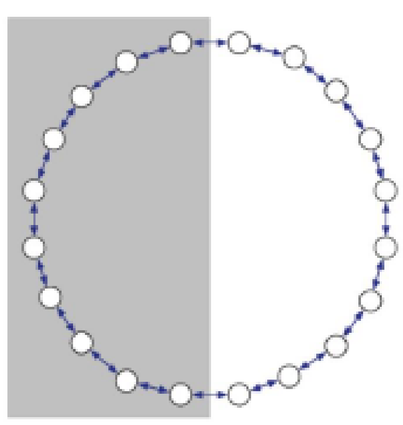

Fig. 3. Half Moon World

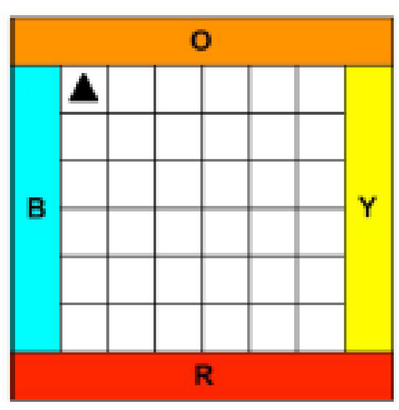

Fig. 4. Grid World

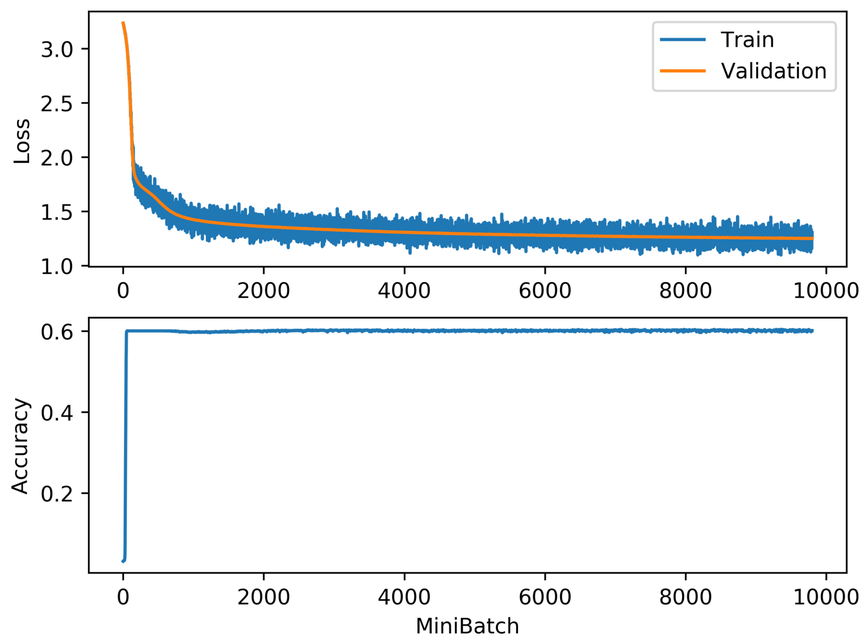

Fig. 5. The loss and accuracy of DNN in Grid world

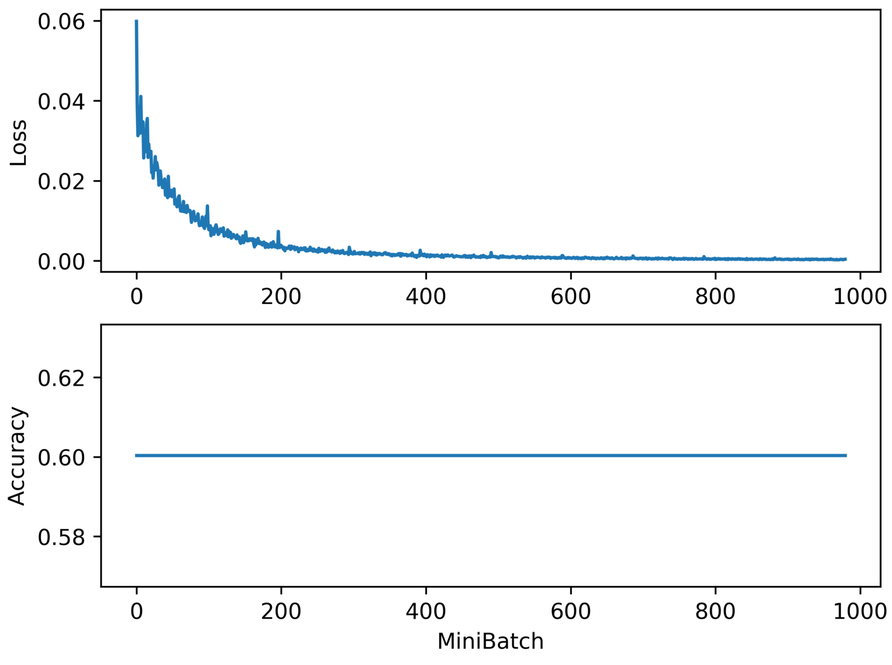

Fig. 6. The loss and accuracy of DPSR-K in Grid world

Table 1. The accuracy of and DNN

Method	Grid world	Half Moon World	Tunnel World
DNN	0.601	0.740	0.517
DPSR-K	0.600	0.654	0.423

Our experiment results show that in two simple task DPSR-K does not receive ideal results, however, for more sophisticated task, our proposed DPSR-K performances better than DNN method. We hypothesis this phenomenon is caused by the lack of state in sample tasks. We will do further research about that in future work.