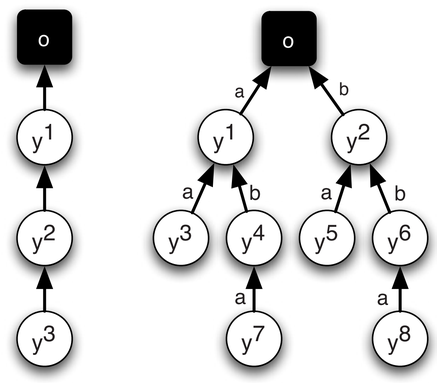
Temporal-Difference Networks for Dynamical Systems with Continuous Observations and Actions

Christopher M. Vigorito
Department of Computer Science
University of Massachusetts Amherst
vigorito@cs.umass.edu

TD Networks

- Temporal-difference (TD) networks[1] are a class of predictive state representation (PSR)[2] which represent state in partially observable dynamical systems in terms of a set of predictions about future observations.
- In a TD network these predictions, or questions, are represented by a question network (see figure below).
- Each node in the network represents a prediction about the value a specific observation will take some number of time steps in the future, possibly conditioned on one or more actions.
- For example, node y^1 below makes a prediction about an observation one time step in the future, either unconditionally (left) or conditioned on action a (right).
- The values of all the nodes in the network at a given time constitute the state of the system, and the values are computed by an answer network.
- The answer network is a function approximator that maps the current predictions, next action, and next observation to a new set of predictions.
- Links in the question network denote node targets, which specify the quantity a node tries to predict. Some targets are in terms of other predictions, and so TD learning techniques can be used to update the parameters of the answer network at each time step.
- Eligibility traces can be used as in traditional TD learning to make more efficient use of data, and in certain systems must be used to guarantee learning.[3]

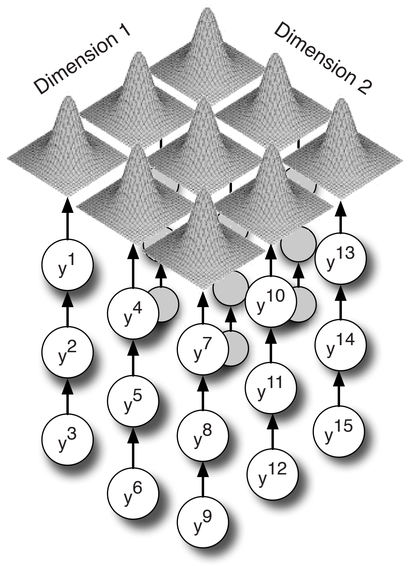


Two example question networks for an uncontrolled system (left) and for a controlled system (right).

- Previous work with TD networks has dealt only with dynamical systems with discrete observations and actions.
- In that work predictions were probabilities of observations taking on specific values from a discrete set.
- The action conditions were binary, although the formalism allowed for non-binary conditions.

Continuous TD Networks

- To extend this work to continuous systems, we consider predicting the expected values of a set of continuously-valued features defined over the observation space of a system.
- The features can be of any form, e.g., radial basis functions, components of a fourier series, etc. The generalization of the network to new observations will thus be based on the form of the features used.
- The figure below represents an example question network for a 2-dimensional uncontrolled system with radial basis functions as features.



References

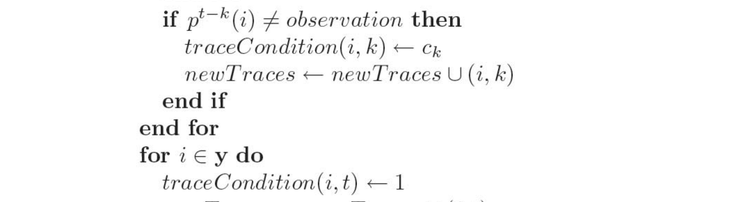
1. Sutton, R. S., & Tanner, B. (2005). Temporal-difference networks. In *Advances in Neural Information Processing Systems* (NIPS) (pp. 1377-1384).
2. Littman, M. L., Sutton, R. S., & Singh, S. (2002). Predictive representations of state. In *Advances In Neural Information Processing Systems* (NIPS) (pp. 1555-1561).
3. Tanner, B., & Sutton, R. S. (2005). Td(λ) networks: temporal-difference networks with eligibility traces. In International Conference on Machine Learning (ICML) (pp. 888-895).
4. Sutton, R. S., & Barto, A. G. (1998). *Reinforcement learning: An introduction.* Cambridge, Massachusetts: MIT Press.

- We must also adapt the existing learning algorithm to handle continuous actions. We assume a set of activation functions that map actions to real numbers in [0,1], each of which represents the similarity of a given action to the point in action space represented by that function.
- For example, in our experiments we use a set of radial basis functions, each centered at a different position in the action space. Each function outputs the degree to which a given action is similar to the point in action space at which that function is centered, based on Euclidian distance. This produces "soft" action conditions that allow for generlization across similar actions.
- Because action conditions are no longer binary, traces are not removed as a result of different actions being taken. Rather, each action condition is multiplied by the value of its associated activation function at each time step, and the weight updates are appropriately scaled by this value.
- Below is the pseudo-code of the TD(λ) algorithm for continuous systems, modified from that for discrete systems as given in Tanner and Sutton (2005).[3] The function traceCondition() here represents the current value of the action condition for a given trace.

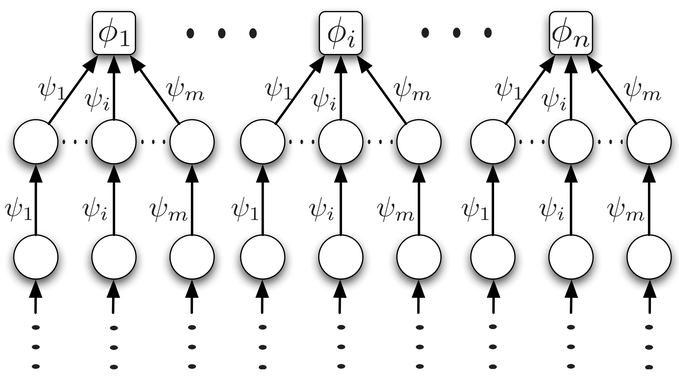
$\Phi \leftarrow$ set of observation feature functions
$\Psi \leftarrow$ set of action activation functions
$\mathbf{y} \leftarrow$ initial state vector
$\mathbf{W} \leftarrow$ initial weight matrix
$Traces \leftarrow \{\}$
for t=0 to T **do**
  $newTraces \leftarrow \{\}$
  $\mathbf{a} \leftarrow chooseAction()$
  $\mathbf{o} \leftarrow getObservation(\mathbf{a})$
  $\mathbf{x}_t \leftarrow x(\mathbf{a}, \mathbf{o}, \mathbf{y}_{t-1}, \Phi, \Psi)$
  $\mathbf{y}_t \leftarrow \mathbf{W}\mathbf{x}_t$
  for $(i,k) \in Traces$ **do**
    if $p^{t-k}(i) \neq observation$ **then**
      $z \leftarrow \mathbf{y}_t[p^{t-k}(i)]$
    else
      $z \leftarrow \phi_{p^{t-k}(i)}(o)$
    end if
    $p \leftarrow \mathbf{y}_{t-1}[p^{t-k-1}(i)]$
    $c_k \leftarrow traceCondition(i,k) \cdot \psi_{p^{t-k-1(i)}}(\mathbf{a})$
    for $w^j \in W[i]$ **do**
      $w^j += \alpha(z-p)c_k x_k^j \lambda^{t-k-1}$
    end for
    if $p^{t-k}(i) \neq observation$ **then**
      $traceCondition(i,k) \leftarrow c_k$
      $newTraces \leftarrow newTraces \cup (i,k)$
    end if
  end for
  for $i \in \mathbf{y}$ **do**
    $traceCondition(i,t) \leftarrow 1$
    $newTraces \leftarrow newTraces \cup (i,t)$
  end for
  $Traces \leftarrow newTraces$
end for

Experiments

- We tested our approach on some simple uncontrolled and controlled, noisy, continuous dynamical systems.
- The uncrontrolled systems were a sinusoid, which output an observation at time t according to (sin(0.5t) + 1)/2, and a square wave, which alternated between outputting 0 and 1 every five time steps.
- Controlled versions of these system were also tested, where the action dimension continuously varied the amplitude of the wave between 0.5 and 1.
- We also tested on a standard reinforcement learning problem, the mountain car[4], but with continuous actions.
- All systems were noisy: mean-zero gaussian noise with variance 0.05 was added to the true observation at each time step.

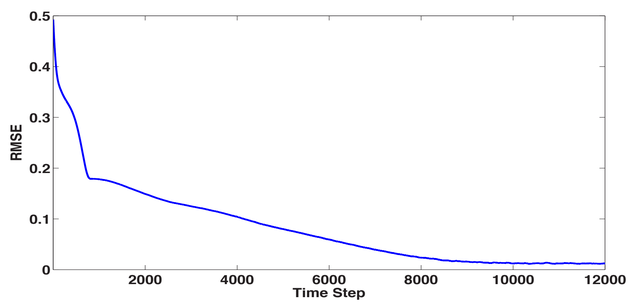


- We employed radial basis functions both for our observation features and our action activation functions. These were tiled evenly over the observation and action spaces. Spherical covariance matrices were used.
- As in previous work we used a linear function approximator to compute the next prediction vector and stochastic gradient descent methods to update the weights of this model.
- Our question networks took the form below (see paper for details). Each φ represents a feature function and each ψ represents an activation function. The depth of the network varied in different experiments.

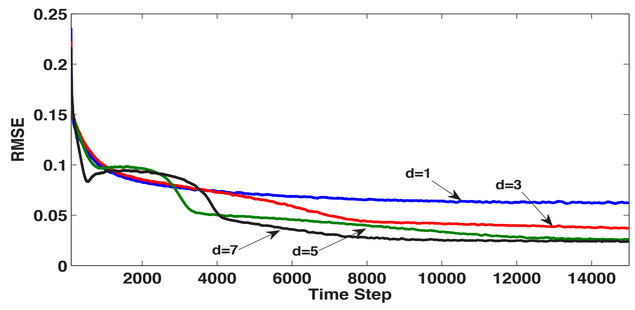


Results

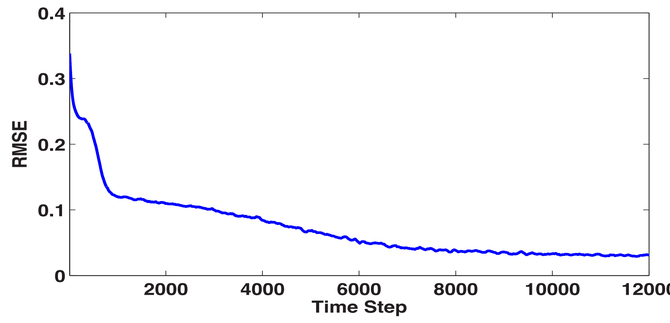
- For each system we calculated the RMSE of the one step predictions of the network, averaged over 30 runs. Each point in each learning curve is an average of the RMSE over the previous 100 points.



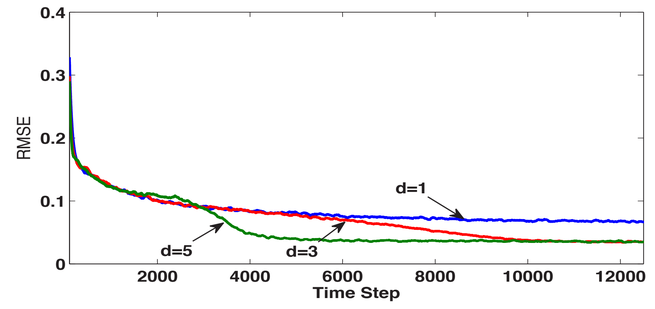
Learning curve for the noisy, uncontrolled square wave.



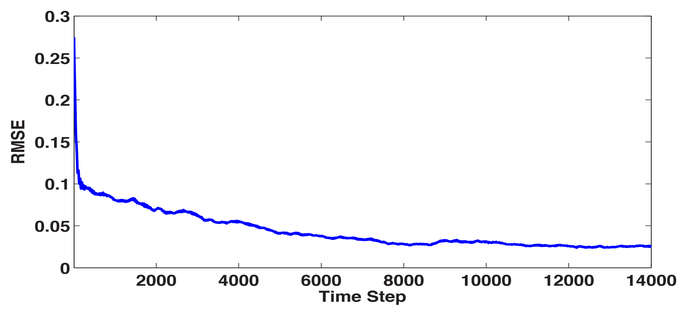
Learning curve for the noisy, uncontrolled sine wave.
Curves for various depths of question network are shown.



Learning curve for the noisy, controlled square wave.



Learning curve for the noisy, controlled sine wave.
Curves for various depths of question network are shown.



Learning curve for the noisy mountain car with continuous actions.

Future Work

- The automatic construction of question networks has been explored for discrete TD networks. We so no reason why that algorithm might not also be used in our setting, though this must be tested empirically.
- Different feature and activation functions for the observation and action spaces will likely produce varying results in the quality of the state approximation. Recent work in basis construction techniques for MDPs may be applied in our setting, allowing these methods to handle partial observability.
- State abstraction has not been explored in TD networks as of yet. In many systems certain dimensions of the observation and action spaces are irrelevant to predicting others. Exploiting this structure may lead to more compact TD network models.