


**Lecture 13**

**Artificial Neural Networks (2 of 4)**

Wednesday, February 16, 2000

Vrushali Koranne  
Department of Computing and Information Sciences, KSU

Readings:  
"TD Models: Modeling the World at a Mixture of Time Scales"  
Richard S. Sutton

CIS 830: Advanced Topics in Artificial Intelligence 

**Presentation Overview**

**Paper**

- "TD Models: Modeling the World at a Mixture of Time Scales"
- Author: Richard S. Sutton, Senior Research Scientist, Computer Sciences Department, University of Massachusetts

**Overview**


- Temporal Difference (TD) Learning
- Intermix TD models at different levels of temporal abstraction (extending beyond a single time step) with a single structure
- TD( $\lambda$ )-style learning algorithm

**Goals**

- TD algorithms for learning multi-scale models for fixed agent behavior


**Reference**

- Reinforcement Learning An Introduction - Richard S. Sutton and Andrew G. Barto

CIS 830: Advanced Topics in Artificial Intelligence 


**Terminology**

- Temporal Difference (TD) Learning
  - Monte Carlo Ideas
    - Learning directly from raw experience without a model of the world's dynamics
  - Dynamic programming
    - Updating estimates based in part on other learned estimates, without waiting for a final outcome
- Markov property
  - State signal that succeeds in retaining all relevant information
  - checker's position
- TD models are used for model-based reinforcement learning architecture in place of conventional Markov models

CIS 830: Advanced Topics in Artificial Intelligence 

**Terminology**

- TD-learning
  - Generalization of the Q-learning
  - more than one-step of lookahead
  - Bootstrapping method: TD method bases its update in part on an existing estimate
  - Sample backups: involves looking ahead to a sample successor state (or state-action pair) using the value of the successor and the reward along the way to compute a backed-up value and then changing value of the original state accordingly
- Features
  - Not just a prediction of rewards but also a prediction of states
  - multi-scale TD models enable planning at higher levels of abstraction
  - Basically a prediction problem - learning a model and value function for the case of fixed agent behavior

CIS 830: Advanced Topics in Artificial Intelligence 


**Presentation Outline**

**Issues**

- Is predicting states the best approach to learn abstract actions or is there a more effective method
- Key strengths - extension beyond a single time step enabled planning at higher level
- Key weaknesses - only prediction problems were considered

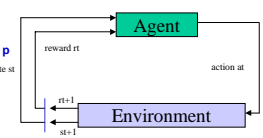
**Outline**

- Overview of Reinforcement Learning
- Prediction Problem and Bellman Equation
- 2 kinds of multi-step models
  - n-step model
  - $\beta$  - model
- Theoretical Results
- TD( $\lambda$ )-style learning algorithm
- Examples
- Future work
- Summary


CIS 830: Advanced Topics in Artificial Intelligence 

**Overview of Reinforcement Learning**

- A learning agent interacting with an environment
- Discrete low level time scale  $t = 0, 1, 2, 3, \dots$
- State  $s_t \in \{1, 2, \dots, m\}$  at time step  $t$
- action  $a_t$  depending on  $s_t$  to change state to  $s_{t+1}$
- $R_t(i)$  - expected value of reward  $r_{t+1}$
- policy  $p$ , a mapping: state  $\rightarrow$  actions
- set of probabilities  $p_a(i, j)$
- discounted rate  $\gamma$ ,  $0 \leq \gamma < 1$
- $V_p(i)$ ; value of state  $i$  under policy  $p$



Agent environment interaction in RL

CIS 830: Advanced Topics in Artificial Intelligence 

## Prediction Problem and Bellman Equation

Prediction Problem	Bellman Equation
<ul style="list-style-type: none"> <li>- estimating the value function <math>V^p</math> for a fixed policy <math>p</math></li> <li>- consider only states and rewards <math>s_0, r_1, s_1, r_2</math></li> <li>- m-vector <math>P</math> of state transition probabilities</li> <li>- m-vector <math>R</math> of expected rewards for each state</li> </ul> $P_{ij} = P(s_j   s_i, a)$ $R_{ij} = R(s_j   s_i, a)$ <p><math>P</math> and <math>R</math> are the 1-step model</p> <ul style="list-style-type: none"> <li>- value function also an m-vector <math>V</math> such that <math>V^T x_i</math> is the value of <math>s_i</math></li> </ul> $V = \sum_{j=0}^{m-1} P_{ij} (R_{ij} + \gamma V_j)$	<ul style="list-style-type: none"> <li>- used to determine the form of models that predict over many time steps</li> <li>- any <math>P</math> and <math>R</math> satisfying the Bellman equation constitute a valid model</li> <li>- thus it can be used to calculate the value function</li> <li>- tells us which temporal details to delete and which to retain</li> <li>- generalized Bellman equation</li> <li>- update and improve an approximation <math>V_i</math> of <math>V</math> by lookahead or backup</li> </ul> $V_i = R_{ij} + \gamma V_j$

CIS 830: Advanced Topics in Artificial Intelligence KSU  
Kansas State University  
Department of Computing and Information Sciences

## Multi-step Models

- n-step Models
  - obtain a 2-step model by expanding the Bellman equation once
$$V_i = R_{ij} + \gamma (R_{jk} + \gamma V_k)$$
- generalize over n steps

**Advantages and Disadvantages**

- requires significantly fewer steps for lookahead for  $V$
- lookahead with different n-step models are completely compatible
- value of  $n$  is fixed for that particular n-step model
- computationally difficult to learn for large values of  $n$

CIS 830: Advanced Topics in Artificial Intelligence KSU  
Kansas State University  
Department of Computing and Information Sciences

## Multi-step Models

**"Blurred" n-step model**

- predicted states occur approximately  $n$  steps later
- $P$  and  $R$  can then hold many different time scales

$$P_{ij} = \sum_{k=0}^{n-1} \beta^k P_{ik} + \beta^n P_{ij}$$

**$\beta$  - Models**

- n-step prediction is given weight  $\beta^n$
- value of  $\beta$  made to vary from state to state
- full  $\beta$  model:  $\beta = 0$  means no weight given; is 1 then state is given all the remaining weight
- provide a very general form of abstract, multi-scale model for stochastic worlds

all weightings must sum to one

CIS 830: Advanced Topics in Artificial Intelligence KSU  
Kansas State University  
Department of Computing and Information Sciences

## Theoretical Results

- Combining the two separate parts of the models into an  $(m+1) \times (m+1)$  matrix
- homogenous co-ordinates can be used for backup as well as forward for multi-step predictions
- Bellman Equation now becomes
- Theorems for Closure under composition, averaging, column wise averaging and Validity of  $\beta$ -models
  - prove validity of all the models discussed earlier in the paper
  - refer to section 8 for proofs

CIS 830: Advanced Topics in Artificial Intelligence KSU  
Kansas State University  
Department of Computing and Information Sciences

## TD( $\lambda$ )-style Learning Algorithm

- Algorithm is for  $\beta$  models
- construct a TD target for next state prediction  $P x_t$
- incremental computation and in learning rate
- Algorithm
  - ideal prediction  $y_t = P x_t$
  - form an n-step TD approximation by replacing each expected state with the observed state
  - next state n-step TD target
  - exponential combination parameterized by  $\lambda$ ,  $0 < \lambda < 1$  (TD  $\lambda$ -style)
  - use the value in an update rule based on the assumption that estimate being updated does not change greatly from time step to time step
  - obtain the learning rule

$$V_t = \sum_{k=0}^{\infty} \gamma^k P x_{t+k}$$

$$V_t = \sum_{k=0}^{\infty} \gamma^k (P x_{t+k} - V_t) + V_t$$

CIS 830: Advanced Topics in Artificial Intelligence KSU  
Kansas State University  
Department of Computing and Information Sciences

## Examples

- Wall-Following Example

**(A) LMS-TD Predictions ( $\gamma=0.9$ )**

**(B) Current Predictions ( $\gamma=0.9$ )**


**(C) Learned Predictions ( $\gamma=0.9$ )**

- application of the  $\beta$  model learning equation
- each attempt treated as a separate trial
- each cell treated as a distinct state, 3 outcomes - colliding, losing-contact, exiting
- distinct states  $\beta = 0$ , all other states  $\beta = 1$
- discount parameter  $\gamma = 0.9$

CIS 830: Advanced Topics in Artificial Intelligence KSU  
Kansas State University  
Department of Computing and Information Sciences

## Examples


- **A Hidden-State Example**
  - overcome hidden state problems
  - 1-step models fail to distinguish between hidden states thus fail in planning
  - $\beta$ -model learned the difference between the two hidden states



$$P_{\text{trans}} = \begin{pmatrix} .84 & .10 & .18 & .47 & .47 & .21 \\ .24 & .69 & .09 & .11 & .13 & .85 \\ .33 & .69 & .09 & .11 & .13 & .85 \\ .86 & .21 & .61 & .69 & .09 & .36 \\ .86 & .60 & .21 & .69 & .09 & .36 \\ .21 & .47 & .47 & .16 & .18 & .84 \end{pmatrix}$$

- states 6 and 7 are ambiguous


CIS 830: Advanced Topics in Artificial Intelligence



## Future Work

- **Future Work**
  - Must extend the pure prediction problem to a reinforcement learning problem which takes into account actions.
  - Author suggests having multiple  $\beta$  models, each corresponding to a different policy i.e. for each type of abstract action
- **Possible improvements**
  - sequence of actions should be taken into consideration
  - choice of actions remains an open question

CIS 830: Advanced Topics in Artificial Intelligence



## Summary

- **Content Critique**
  - **Key Contribution** - predicting states rather than predicting rewards
  - **Strengths**
    - Approach to multi-level modeling and planning applying to a broad class of stochastic processes
    - each high-level model has a clear definitive semantics
    - provides a good mathematical base to explain the modeling process
    - positive step towards making ANNs learn abstract actions
  - **Weaknesses**
    - focus is exclusively on temporal aspects of abstraction, each state bearing no relation with other states
    - ignores actions and deals with prediction problems rather than control of problems
    - how far can we generalize this approach
- **Presentation Critique**
  - Audience - AI, Robotics (making neural networks learn abstract actions)
  - Positive points - Good mathematical foundation and substantiated by examples
  - Negative points - more stress on proving theories

CIS 830: Advanced Topics in Artificial Intelligence

