Multi-step Dyna-style Planning

Hengshuai Yao,
Joint work with Rich Sutton, Shalabh Bhatnagar,
Dongcui Diao, Csaba Szepesvari

Workshop of Abstraction in RL

18 June, 2009, Montreal
Outline

- Planning, Dyna, *Multi-step Dyna Planning*
- Linear multi-step Model
- Planning using linear, multi-step model
- Some experimental results
Planning

- Model-based RL; “Any computational process that takes a model as input and produces or improves a policy for interacting with the modeled environment.” (Sutton and Barto, 98)

- Examples: DP, Heuristic Search, Dyna
Dyna Planning Architecture

Dyna Agent

Real Experience

Simulated Experience

Model

Value Funcion

Planning

observation

action
Multi-step Dyna planning?

- is an integrate Dyna architecture of planning using a *multi-step model* of the world

- Models: state model, state-action model
Multi-step Models

• is a world model that tells the future results (many steps later) given input of the model.

• can be tabular,
  • can be one-step, e.g., the Bellman equation
  • can be multi-step, e.g., the k-step model by Sutton 95

• can be linear, e.g., the one-step model by (Sutton, Szepesvari, Geramifard, Bowling 08)

• can be linear, multi-step, ....
Single-step/Multi-step Models

One-step Linear Model

Expected Next Feature

Approximated One-step Reward

(Sutton et al., 08)

k-step Linear Model

Expected n-step Feature

Approximated n-step Reward
Single-step Models

Single-step tabular model

\[ P_{i,j}^{\pi} = E_\pi \{ s_{t+1} = j \mid s_t = i \} \]

\( R^{\pi} \)  is the expected reward of leaving state \( i \) in one step

Single-step linear model (Sutton et al., 08)

\[
(F^{\pi})' = (\Phi' D^{\pi} \Phi)^{-1} \cdot \Phi' D^{\pi} P^{\pi} \Phi, \\

f^{\pi} = (\Phi' D^{\pi} \Phi)^{-1} \cdot \Phi' D^{\pi} R^{\pi}
\]
Multi-step Linear Model

- Properties

k-step Linear Model

- When LFA reduces to tabular, reducing to the n-step model

\[ F^{(k)} = (\gamma F^\pi)^k, \quad f^{(k)} = \sum_{j=0}^{k-1} (\gamma (F^\pi)')^j f^\pi \]

- n=1, reducing to current one-step linear model

k-step tabular model (Sutton 95)

- Well defined or any n>1, and has the same solution with TD

\[ P^{(k)} = (\gamma (P^\pi)')^k, \quad R^{(k)} = \sum_{j=0}^{k-1} (\gamma P^\pi)^j R^\pi, \quad k = 1, 2, \ldots \]
Set $\tilde{\theta}_0 = \theta_{t+1}$

for $\tau = 1$ to $p$ do
  Sample $\tilde{\phi}_\tau \sim \mu(\cdot)$
  $\tilde{\phi}^{(k)} = F^{(k)} \tilde{\phi}_\tau$
  $\tilde{r}^{(k)} = (f^{(k)})' \tilde{\phi}_\tau$
  $\tilde{\theta}_{\tau+1} = \tilde{\theta}_\tau + \alpha_\tau (\tilde{r}^{(k)} + \tilde{\theta}'_\tau \tilde{\phi}^{(k)} - \tilde{\theta}'_\tau \tilde{\phi}_\tau) \tilde{\phi}_\tau$
end for

Set $\theta_{t+1} = \tilde{\theta}_{\tau+1}$
Results: Boyan Chain

- Decoupling of learning, modeling, planning
- Using \textit{lambda-model}, efficient to estimate
Conclusion

- Linear Dyna is constantly faster than TD learning: (stronger than Sutton, et., al., 08)
- Multi-step Dyna is faster than linear Dyna
- Planning for control
  - builds a multi-step, state-action model
  - tracks the greedy policy

For details, visit our poster :)