1 Planning and Learning with Tabular Methods

- model-based: rely on *planning*; require a model of the environment (anything that an agent can use to predict how the environment will respond to its actions), e.g. dynamic programming and heuristic search
- model-free: rely on *learning*; use without a model, such as Monte Carlo and temporal-difference method

1.1 Models and Planning

- distribution model: models that produce a description of all possibilities and their probabilities (e.g. MDP dynamics $p(s', r|s, a)$
- sample models: models that produce just one of the possibilities, sampled according to the probabilities (e.g. HTHTHH flipping coin sequence)

Distribution models are stronger than sample models in that they can always be used to produce samples; however, sample models are easier to implement.

Definition 1.1 ***planning****, refer to any computational process that takes a* ***model*** *as input and produces or* ***improves a policy*** *for interacting with the modeled environment.*

$$\textit{model} \xrightarrow{\textit{planning}} \textit{policy}$$

- state-space planning: a search through the state space for an π_*, or an optimal path to a goal
- plan-space planning: a search through the space of plans
  - includes evolutionary methods, and partial-order planning (ordering of steps is not completely determined at all stages of planning)
  - hard to apply to the stochastic sequential decision problems
  - not focused in this book

State-space planning methods' structure: (1) all state-space planning methods involve computing value functions to improve the policy, (2) they compute value functions by updates or backup operations applied to simulated experi-

ence.

$$\text{model} \rightarrow \text{simul. exp.} \rightarrow \xrightarrow{\text{backup}} \rightarrow \text{values} \rightarrow \text{policy}$$

State-space planning methods fits in the above structure, only differed by (1) update rule, (2) order of update, (3) how long the backed-up info is retained.

The heart of both planning and learning methods is the estimation of value functions by backing-up update operations.

- planning uses simulated experience generated by a model (e.g. DP)
- learning uses real experience generated by the environment (e.g. TD)

Random-sample one-step tabular Q-planning

Loop forever:

1. Select a state, $S \in \S$, and an action, $A \in \mathcal{A}(\S)$, at random
2. Send S, A to a sample model, and obtain a sample next R, and S'

Apply one-step tabular **Q-learning** to S, A, R, S':

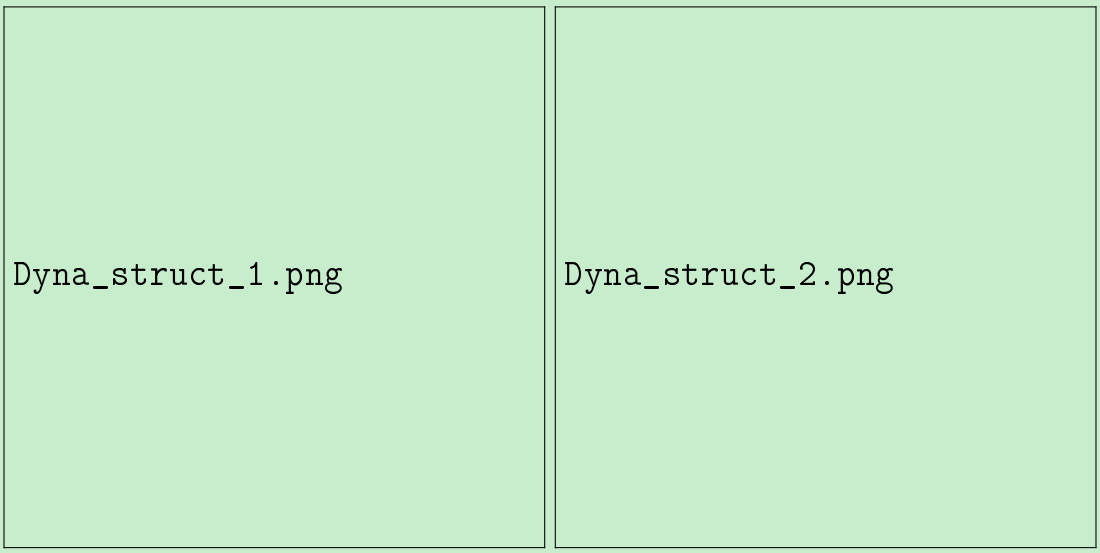
$$Q(S,A) \leftarrow Q(S,A) + \alpha[R + \gamma max_a Q(S',a) - Q(S,A)]$$

1.2 Dyna: Integrated Planning, Acting, and Learning

Problem: Both decision making and model learning are computation-intensive

Solution: To balance these two, we use a architecture to integrate the major functions in an online planning agent, called Dyna-Q.

- real experience: (1) *model-learning* (or indirect RL) to improve model (more accurately match the real environment); (2) *direct RL* to improve value function and policy
  - indirect methods: maximize the use of limited experience (better policy with fewer environmental interactions)
  - direct methods are more simpler and are not affected by biases incurred by the model
- all planning, acting, model-learning, and direct RL occurring continually

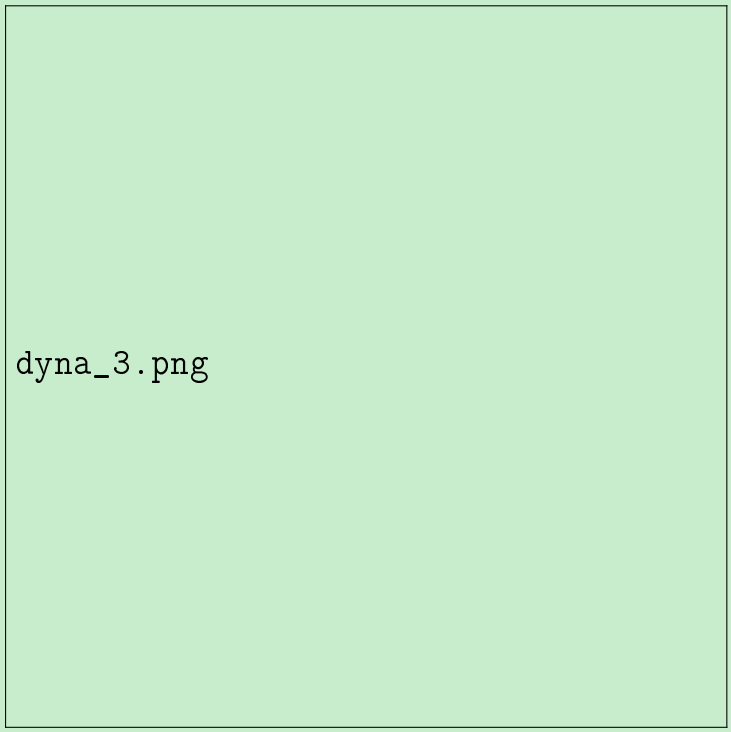


(a) (b)

Figure 1: (a) Dyna, balancing decision making and model learning (b) General Dyna Architecture

- planning here uses random-sample one-step tabular Q-**planning**
- direct RL here uses one-step tabular Q-**learning**

- *search control* to refer to the process that selects the starting states and actions for the simulated experiences generated by the model



The Dyna Q uses both real world experience (which is expensive) and sim-

ulated (hallucinated) experience (which is cheap; more iterations are completed with simulated experience), thus accelerating training. The real experience is for learning, and simulated experience is for planning.

Tabular Dyna-Q

Init $Q(s,a)$ and $Model(s,a)$ for all $s \in \S$ and $a \in \mathcal{A}(s)$

Loop forever:

- $S \leftarrow$ current (nonterminal) state
- $A \leftarrow \epsilon$-greedy(S,Q)
- Take action A; observe resultant reward, R, and state, S'
- $Q(S,A) \leftarrow Q(S,A) + \alpha[R + \gamma max_a Q(S',a) - Q(S,A)]$
- $Model(S,A) \leftarrow R, S'$ (assuming deterministic environment)
- Loop repeat n times:
  - $S \leftarrow$ random previously observed state
  - $A \leftarrow$ random action previously taken in S
  - $R, S' \leftarrow Model(S,A)$
  - $Q(S,A) \leftarrow Q(S,A) + \alpha[R + \gamma max_a Q(S',a) - Q(S,A)]$

1.3 When the Model Is Wrong

The model learning in the Dyna-Q may be incorrect when encountering stochastic environment. The model may not be exploratory enough to find the new (optimal) path when environment changes, sticking to the old path. We slightly modified the reward transition in Dyna-Q+ with $r + \kappa\sqrt{\tau}$ for small κ, where τ is the time steps that the (s,a) has not been tried (similar to how UCB incorporate time steps inside).

1.4 Learning Objectives (UA RL MOOC)

Lesson 1: What is a model?

1. Describe what a model is and how they can be used

- Model are used to store knowledge about the transition and reward dynamics
- Given S, A into model, model outputs R, S'
- A model allows for planning
- Planning refers to the process of using a model to improve a policy (use model to simulate experience, update value functions with the simulated experience, then improve policy with the updated value functions)

2. Classify models as distribution models or sample models

- Sample models procedurally generate samples, without explicitly storing the probability of each outcome (e.g. flip coin twice: HT)
- Distribution models contain a list of all outcomes and their probabilities (e.g. flip coin twice HT(1/4),HH(1/4),TT(1/4),TH(1/4))

3. Identify when to use a distribution model or sample model

Sample model can be computationally inexpensive. Distribution model contains more info, but it's hard to specify and can become large.

4. Describe the advantages and disadvantages of sample models and distribution models

Sample models require less memory

Distribution models can be used to compute the exact expected outcome (note sample models have to averaging many samples to get an approximate). Can be used to access risk.

5. Explain why sample models can be represented more compactly than distribution models

Consider rolling dice sample. The more dice there are, the larger the state space.

Lesson 2: Planning

6. Explain how planning is used to improve policies

Planning uses simulated experience from model to improve policies.

7. Describe random-sample one-step tabular Q-planning

(1) sample from model; (2) Q-learning update; (3) Greedy policy improvement

Lesson 3: Dyna as a formalism for planning

8. Recognize that direct RL updates use experience from the environment to improve a policy or value function

Direct RL, like Q-learning, directly learn from real world experience (environment)

9. Recognize that planning updates use experience from a model to improve a policy or value function

Indirect RL, like Q-planning, learn from simulated experience (generated by model) to improve value function and policy.

10. Describe how both direct RL and planning updates can be combined through the Dyna architecture

11. Describe the Tabular Dyna-Q algorithm

12. Identify the direct-RL and planning updates in Tabular

13. Identify the model learning and search control components of Tabular Dyna-Q

10-13 see chap 8.2

14. Describe how learning from both direct and simulated experience impacts performance

It accelerate learning

15. Describe how simulated experience can be useful when the model is accurate

Lesson 4: Dealing with inaccurate models

16. Identify ways in which models can be inaccurate

17. Explain the effects of planning with an inaccurate model

18. Describe how Dyna can plan successfully with a partially inaccurate model

19. Explain how model inaccuracies produce another exploration-exploitation trade-off

20. Describe how Dyna-Q+ proposes a way to address this trade-off