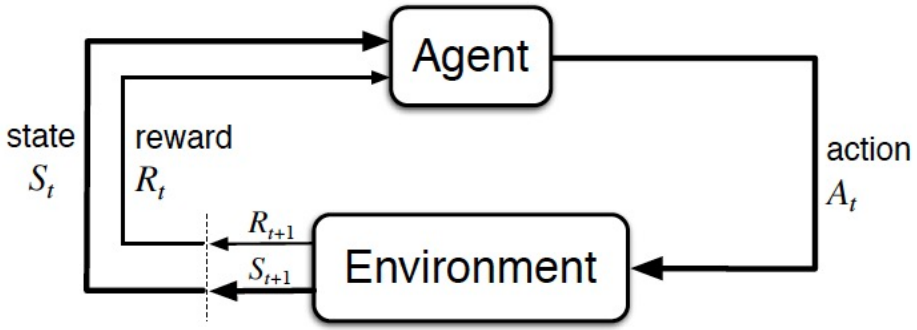
Remarks on MDPs & Dynamic Programming

Christopher Mutschler

Recap: Intro to Reinforcement Learning

- The RL Paradigm (revisited):
  - Do you agree with following statement?

"All goals can be described by the maximization of expected cumulative reward."

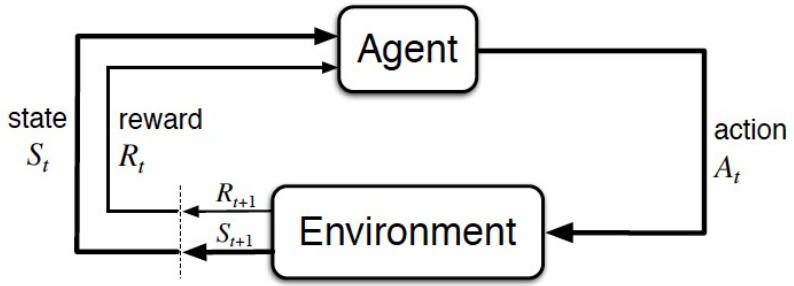


Sutton, R. S., & Barto, A. G. (2018). Reinforcement learning: An introduction. MIT press.

thanks to @karpathy

Markov Decision Processes

- Agent learns by interacting with an environment over many time-steps:
- Markov Decision Process (MDP) is a tool to formulate RL problems
  - Description of an MDP $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$:



Sutton, R. S., & Barto, A. G. (2018). Reinforcement learning: An introduction. MIT press.

Note:
If the interaction does stop at some point in time (T) then we have an *episodic RL problem.*

- At each step t, the agent:
  - is at state S_t,
  - performs action A_t,
  - receives reward R_t.
- At each step t, the environment:
  - receives action A_t from the agent,
  - provides reward R_t,
  - moves at state S_{t+1},
  - increments time $t \leftarrow t + 1$.

Markov Decision Processes

- Expected long-term value of state s:

$$v(s) = \mathbb{E}(\mathrm{G}) = \mathbb{E}(R_0 + \gamma R_1 + \gamma^2 R_2 + \gamma^3 R_3 + \ldots + \gamma^t R_t)$$

- Goal: maximize the expected return $\mathbb{E}(\boldsymbol{G})$.

- We need a controller that helps us select the actions to maximize $\mathbb{E}(G)$.
- A policy π represents this controller:
  - π determines the agent's behavior, i.e., its way of acting
  - π is a mapping from state space $\mathcal{S}$ to action space $\mathcal{A}$

$$\pi : \mathcal{S} \mapsto \mathcal{A}$$

  - Two types of policies:
    - Deterministic policy: $a = \pi(s)$.
    - Stochastic policy: $\pi(a \mid s) = \mathbb{P}[A_t = a \mid S_t = s]$.

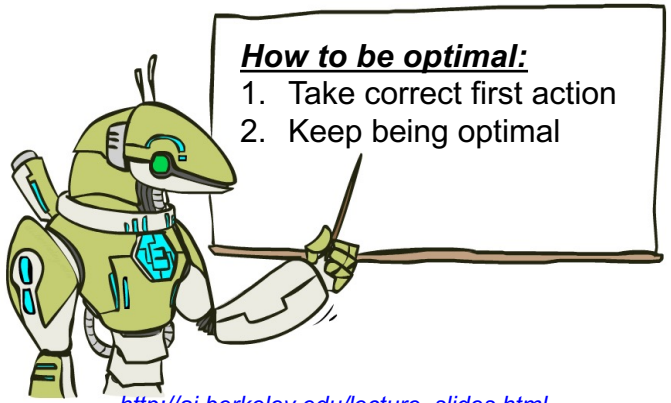
- New goal: find a policy that maximizes the expected return!

Dynamic Programming

- How do we find optimal controllers for given (known) MDPs?
- Bellman equation & Bellman's principle of optimality

Principle of Optimality:
„An optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision."

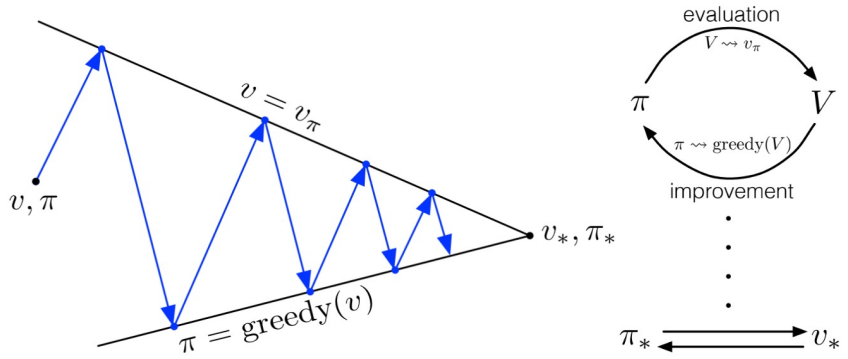
(see Bellman, 1957, Chap. III.3.)



http://ai.berkeley.edu/lecture_slides.html

Dynamic Programming

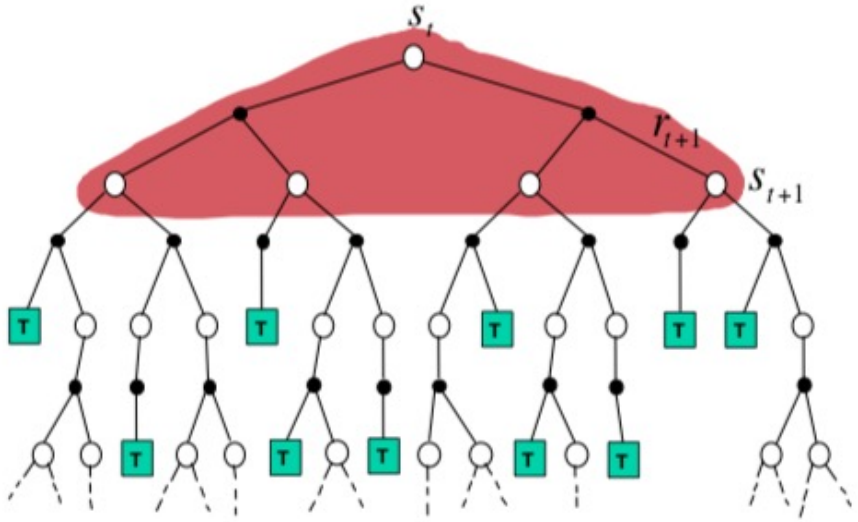
- Dynamic Programming (DP) methods to find optimal controllers
  - DP methods are guaranteed to find optimal solutions for Q and V in polynomial time (in number of states and actions) and are exponentially faster than direct search
  - Policy Iteration computes the value function under a given policy to improve the policy while value iteration directly works on the states
    - Perform sweeps through the state set
    - Implement the Bellman equation update
    - Use bootstrapping



Sutton, R. S., & Barto, A. G. (2018). Reinforcement learning: An introduction. MIT press.

Dynamic Programming

- Backup Diagrams



Sutton, R. S., & Barto, A. G. (2018). Reinforcement learning: An introduction. MIT press.

Dynamic Programming

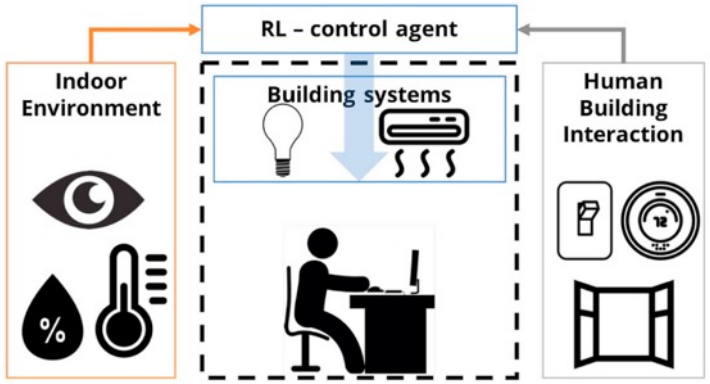
- **But are these simple algorithms usable?**

Dynamic Programming

- **But are these simple algorithms usable?**
- LightLearn: personalized lighting control with Value Iteration and learned transition model

Table 3
State definitions and rewards (note that P_4 is defined as both early morning and late night).

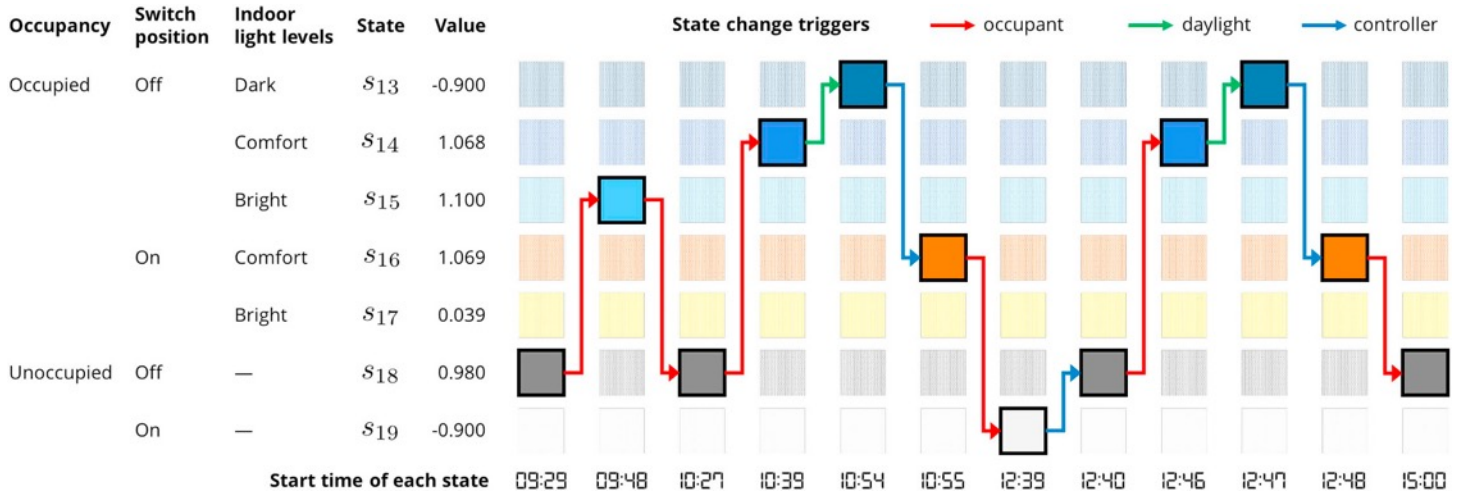
Occupancy	Switch position	Indoor light levels	Period of day				Reward
			P_4	P_1	P_2	P_3	
Occupied	Off	Dark	s_1	s_6	s_{13}	s_{20}	− 1
		Comfort	s_2	s_7	s_{14}	s_{21}	+ 1
		Bright	–	s_8	s_{15}	s_{22}	+ 1
	On	Comfort	s_3	s_9	s_{16}	s_{23}	+ 1
		Bright	–	s_{10}	s_{17}	s_{24}	0
Unoccupied	Off	–	s_4	s_{11}	s_{18}	s_{25}	+ 1
	On	–	s_5	s_{12}	s_{19}	s_{26}	− 1



Park, J. Y., Dougherty, T., Fritz, H., & Nagy, Z. (2019). LightLearn: An adaptive and occupant centered controller for lighting based on reinforcement learning. *Building and Environment*, *147*, 397-414.

Dynamic Programming

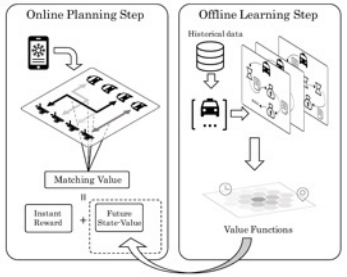
- **But are these simple algorithms usable?**
- LightLearn: personalized lighting control with Value Iteration and learned transition model



Park, J. Y., Dougherty, T., Fritz, H., & Nagy, Z. (2019). LightLearn: An adaptive and occupant centered controller for lighting based on reinforcement learning. *Building and Environment*, *147*, 397-414.

Dynamic Programming

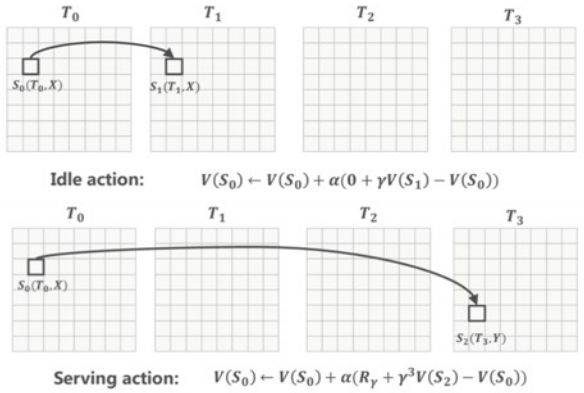
- **But are these simple algorithms usable?**
- Large-scale order dispatch in on-demand ride-hailing platforms
  - Problem: find the best matching between drivers and orders (e.g. Uber)
  - Available information: each taxi uploads occupancy status and location in central platform
  - Classical solution: during each short time slot (say one or two seconds), the platform's decision center first collects all the available drivers and active orders, and then matching is based on a combinatorial optimization algorithm.

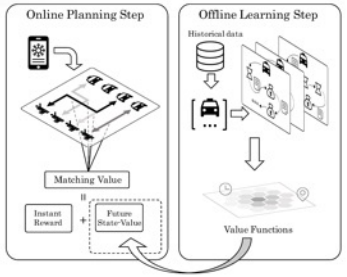


Xu, Z., Li, Z., Guan, Q., Zhang, D., Li, Q., Nan, J., ... & Ye, J. (2018, July). Large-scale order dispatch in on-demand ride-hailing platforms: A learning and planning approach. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (pp. 905-913). ACM.

Dynamic Programming

- **But are these simple algorithms usable?**
- Large-scale order dispatch in on-demand ride-hailing platforms
  - Idea: design an order dispatch algorithm that optimizes the platform's global efficiency in a long horizon (e.g., two or three hours or a day), by formulating order dispatch as a large-scale sequential decision-making problem





Xu, Z., Li, Z., Guan, Q., Zhang, D., Li, Q., Nan, J., ... & Ye, J. (2018, July). Large-scale order dispatch in on-demand ride-hailing platforms: A learning and planning approach. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (pp. 905-913). ACM.

Dynamic Programming

- **But are these simple algorithms usable?**
- Large-scale order dispatch in on-demand ride-hailing platforms
  - Reward: the price of an order → Goal: maximize the Gross Merchandise Volume for the entire platform
  - Goal: Online planning: takes the learned value functions as inputs and determines the final matching between drivers and orders in real-time

Xu, Z., Li, Z., Guan, Q., Zhang, D., Li, Q., Nan, J., ... & Ye, J. (2018, July). Large-scale order dispatch in on-demand ride-hailing platforms: A learning and planning approach. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (pp. 905-913). ACM.

References

- Books:
  - Sutton, R. S., & Barto, A. G. (2018). Reinforcement learning: An introduction. MIT press.
  - Bellman, R.E. 1957. Dynamic Programming. Princeton University Press, Princeton, NJ. Republished 2003: Dover, ISBN 0-486-42809-5.

- Lectures:
  - UC Berkeley CS188 Intro to AI. http://ai.berkeley.edu/lecture_slides.html
  - UCL Course on RL. http://www0.cs.ucl.ac.uk/staff/d.silver/web/Teaching.html
  - Advanced Deep Learning and Reinforcement Learning (UCL + DeepMind). http://www.cs.ucl.ac.uk/current_students/syllabus/compgi/compgi22_advanced_deep_learning_and_reinforcement_learning

- Blogs etc.:
  - https://cs.stanford.edu/people/karpathy/reinforcejs/gridworld_dp.html