Lecture 15

# Markov Decision Processes

CS 3630

## *Markov Chains: Recap*

# Controlled Markov Chains

Depending on context, we sometimes use $\boldsymbol{a_t}$ and sometimes use $\boldsymbol{u_t}$.

A controlled Markov chain is defined by

- A set of states: $\boldsymbol{\mathcal{X}}$
- A set of actions: $\boldsymbol{\mathcal{A}}$
- A set of transition probabilities: $\boldsymbol{P(X_{t+1} = x_{t+1} | \, X_t = x_t, a_t \,)}$

The Markov chain is the sequence of random states: $\boldsymbol{X_0, X_1, \ldots, X_n}$

The key idea behind Markov chains is that the past and future are completely decoupled if we know the current state. This can be written mathematically as:

$$\boldsymbol{P(X_{t+1} = x_{t+1} \mid a_0, \ldots a_t, X_0 = x_0, \ldots, X_t = x_t) = P(X_{t+1} = x_{t+1} | \; X_t = x_t, a_t)}$$

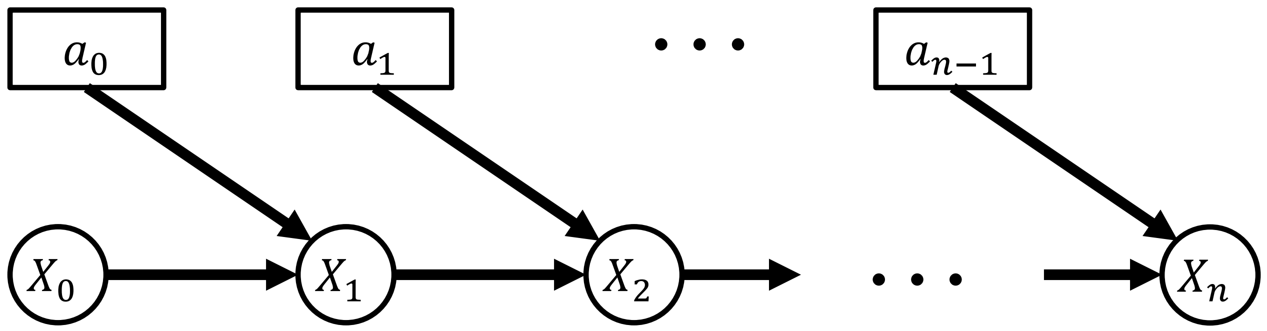
The belief state $\boldsymbol{b_{t+1}}$ is a vector that specifies $\boldsymbol{P(X_{t+1} = x_{t+1} \mid a_0, \ldots a_t)}$ for each possible $\boldsymbol{x_{t+1}}$.

# Controlled Markov Chains

- The action $a_t$ does not determine the next state!
- It determines which conditional probability distribution will be used to determine the posterior for $X_{t+1}$.
- Since we know $a_1 \dots a_t$, we can compute the posterior $P(X_{t+1} = x_{t+1} | a_1 \dots a_t)$ by working forward from the initial distribution $P(X_0)$ using

$$P(X_{t+1} = x_{t+1} | a_1 \dots a_t) = \sum_{x_t} P(X_{t+1} = x_{t+1} | X_t = x_t, a_t) P(X_t = x_t | a_1, \dots a_{t-1})$$

- Controlled Markov chains have a nice graphical representation:



Note:
***States*** are ***random*** – circles.
***Actions*** are ***deterministic*** – boxes.

# State transitions and the belief state

| XT | AT | Living Room | Kitchen | Office | Hallway | Dining Room |
|---|---|---|---|---|---|---|
| Living Room | L | 1 | 0 | 0 | 0 | 0 |
| Living Room | R | 0.2 | 0.8 | 0 | 0 | 0 |
| Living Room | U | 1 | 0 | 0 | 0 | 0 |
| Living Room | D | 0.2 | 0 | 0 | 0.8 | 0 |
| Kitchen | L | 0.8 | 0.2 | 0 | 0 | 0 |
| Kitchen | R | 0 | 1 | 0 | 0 | 0 |
| Kitchen | U | 0 | 1 | 0 | 0 | 0 |
| Kitchen | D | 0.2 | 0 | 0 | 0 | 0.8 |
| Office | L | 0 | 0 | 1 | 0 | 0 |
| Office | R | 0 | 0 | 0.2 | 0.8 | 0 |
| Office | U | 0 | 0 | 1 | 0 | 0 |
| Office | D | 0 | 0 | 1 | 0 | 0 |
| Hallway | L | 0 | 0 | 0.8 | 0.2 | 0 |
| Hallway | R | 0 | 0 | 0 | 0.2 | 0.8 |
| Hallway | U | 0.8 | 0 | 0 | 0.2 | 0 |
| Hallway | D | 0 | 0 | 0 | 1 | 0 |
| Dining Room | L | 0 | 0 | 0 | 0.8 | 0.2 |
| Dining Room | R | 0 | 0 | 0 | 0 | 1 |
| Dining Room | U | 0 | 0.8 | 0 | 0 | 0.2 |
| Dining Room | D | 0 | 0 | 0 | 0 | 1 |

- Consider the action move right.
- We construct the conditional probability matrix for this action by collecting the move right rows from the table.

$$M_r = \begin{bmatrix} 0.2 & 0.8 & 0.0 & 0.0 & 0.0 \\ 0.0 & 1.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.2 & 0.8 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.2 & 0.8 \\ 0.0 & 0.0 & 0.0 & 0.0 & 1.0 \end{bmatrix}$$

We can define a transition matrix for each action using the conditional probability table:

$$b_{t+1} = b_t M_r$$

Given the transition probabilities $P(X_{t+1} = x_{t+1} | \, X_t = x_t, a_t)$, the action sequence $a_0, \ldots, a_t$, and the prior $P(X_t)$, we can compute the belief state for any future time.

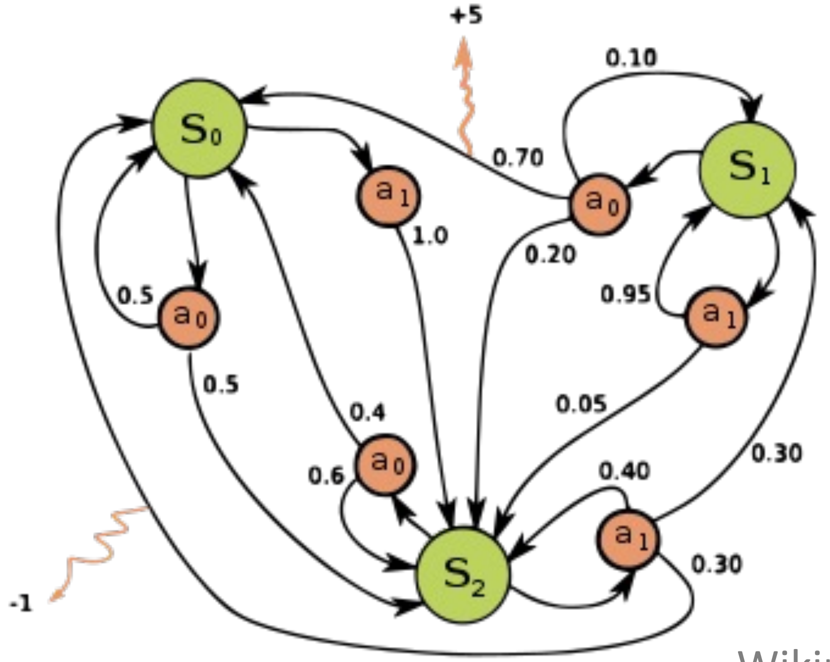
## Markov Decision Processes

- We have seen controlled Markov chains that are driven by a sequence of actions, $a_0, \dots a_n$.
- **Planning** is the process of choosing which actions to perform.
- In order to plan effectively, we need quantitative criteria to evaluate actions and their effects.
- MDPs include a reward function that characterizes the immediate benefit of applying an action.
- Policies describe how to act in a given state.
- The value function characterizes the long-term benefits of a policy.
- We assume that the robot is able to ***know*** its current state with certainty.

➢ ***We will see how to define reward functions and use these to compute optimal policies for MDPs.***

# Markov decision process (MDP)

- A Markov decision process (MDP) is a stochastic decision-making process
  - a mathematical framework that models the decision-making of a dynamic system in scenarios where the results are either random or controlled by a sequential decision making process
- MDPs are defined by:
  - X – a set of states
  - A – a set of actions actions
  - $P(x, a, x')$ - transition probability fn
  - $R(x, a, x')$ - reward fn



Wikipedia

## Reward Functions

A ***reward function*** gives a quantitative evaluation of the benefit of an action with respect to a state transition.

- The most general form depends on current state, action, and next state:

$$R: \mathcal{X} \times \mathcal{A} \times \mathcal{X} \to \mathbb{R}$$

- $R(x_t, a_t, x_{t+1}) =$ Reward for transitioning from state $x_t$ to state $x_{t+1}$ by executing action $a_t$.

- *Note that $x_t, a_t, x_{t+1}$ are fully specified. We can only evaluate the reward in retrospect!*

- *Example:*

  $R(H, up, L)$*= reward for moving up from the hallway to the living room using the "move up" action.*

# More About Reward Functions

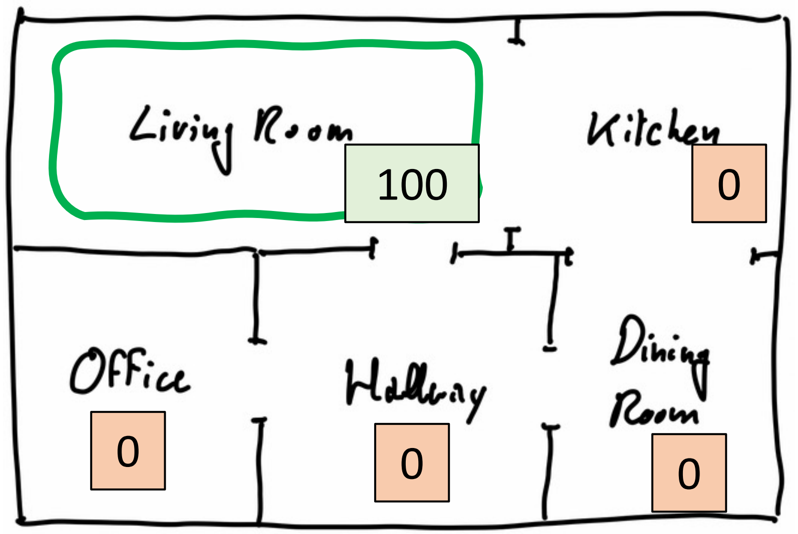
- The reward function is typically time invariant, and can be written as: $R(x, a, x')$.
- The form $R(x, a, x')$ is very general.

Present state  Action  Next state

- Today, we'll use a simplified form: $R(x')$
  - This reward *depends only on destination state*, and *does not* depend on the present state or on the action that is executed.
  - For example, if the goal is to reach the living room, we might have:

$$R(L) = 100,$$
$$R(K) = R(H) = R(O) = R(D) = 0$$

  - In this case there is no reward until the goal is reached.



# Expected Reward

- We define the ***expected reward*** for executing action $a$ in state $x$ as: $\bar{R}(x,a) = E[R(x,a,X')]$

$X'$ is the only random quantity

- Since we know the transition probabilities, we can easily compute this:

$$\bar{R}(x,a) = E[R(x,a,X')] = \sum_{x'} P(x'|x,a) R(x,a,x')$$

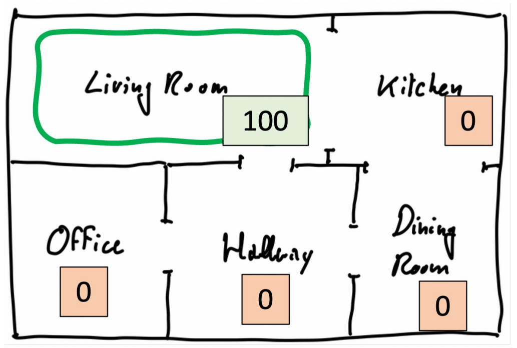
- A simple, greedy planning algorithm is to merely choose the action that maximizes the immediate reward.
  For state $x$, choose the action that maximizes the expected reward:

$$a^* = \arg\max \bar{R}(x,a)$$

# Example

- The expected immediate reward for all four actions in the Kitchen:

$$\bar{R}(K,L) = 80$$
$$\bar{R}(K,R) = 0$$
$$\bar{R}(K,U) = 0$$
$$\bar{R}(K,D) = 20$$



- Hence, when in the kitchen, always do L !
- This is a greedy planning algorithm

| *X1* | *A1* | Living Room | Kitchen | Office | Hallway | Dining Room |
|---|---|---|---|---|---|---|
| **Living Room** | **L** | 1 | 0 | 0 | 0 | 0 |
| **Living Room** | **R** | 0.2 | 0.8 | 0 | 0 | 0 |
| **Living Room** | **U** | 1 | 0 | 0 | 0 | 0 |
| **Living Room** | **D** | 0.2 | 0 | 0 | 0.8 | 0 |
| **Kitchen** | **L** | 0.8 | 0.2 | 0 | 0 | 0 |
| **Kitchen** | **R** | 0 | 1 | 0 | 0 | 0 |
| **Kitchen** | **U** | 0 | 1 | 0 | 0 | 0 |
| **Kitchen** | **D** | 0.2 | 0 | 0 | 0 | 0.8 |
| **Office** | **L** | 0 | 0 | 1 | 0 | 0 |
| **Office** | **R** | 0 | 0 | 0.2 | 0.8 | 0 |
| **Office** | **U** | 0 | 0 | 1 | 0 | 0 |
| **Office** | **D** | 0 | 0 | 1 | 0 | 0 |
| **Hallway** | **L** | 0 | 0 | 0.8 | 0.2 | 0 |
| **Hallway** | **R** | 0 | 0 | 0 | 0.2 | 0.8 |
| **Hallway** | **U** | 0.8 | 0 | 0 | 0.2 | 0 |
| **Hallway** | **D** | 0 | 0 | 0 | 1 | 0 |
| **Dining Room** | **L** | 0 | 0 | 0 | 0.8 | 0.2 |
| **Dining Room** | **R** | 0 | 0 | 0 | 0 | 1 |
| **Dining Room** | **U** | 0 | 0.8 | 0 | 0 | 0.2 |
| **Dining Room** | **D** | 0 | 0 | 0 | 0 | 1 |

$$\bar{R}(K,L) = \sum_{x'} P(x'|x,a)R(x,a,x') = (0.8\times 100) + (0.2\times 0) + (0\times 0) + (0\times 0) = 80$$

# Utility

*Suppose we take a sequence of actions* $A_{1:n} = (a_1, \ldots, a_n)$ *and as a result, we visit a sequence of states:* $X_{1:n+1} = (x_1, \ldots, x_n, x_{n+1})$.

*What should be the accumulated reward??*

- Introduce a discount factor $\gamma$ to
    - Bias towards more immediate payoff
    - Discount the reward for actions far into the future
    - Allow infinite time horizons

*We define the Utility,* $U\colon \mathcal{A}^n \times \mathcal{X}^{n+1} \rightarrow \mathbb{R}$ by:

$$U(a_1, \ldots, a_n, x_1, \ldots x_{n+1}) = R(x_1, a_1, x_2) + \gamma R(x_2, a_2, x_3) + \cdots \gamma^{n-1} R(x_n, a_n, x_{n+1})$$

➢*NOTE: There is* ***nothing*** *random on this slide!*

# Discount Factor

We can write the utility as a sum:

$$U\left(a_{1,} \ldots, a_n, x_1, \ldots x_{n+1}\right) = \sum_{i=1}^{n} \gamma^{i-1} R(x_i, a_i, x_{i+1})$$

- The value of $n$ determines the problem horizon.
- If we let $n \rightarrow \infty$ we have an ***infinite horizon*** problem.
- For $n < \infty$, we have a ***finite horizon*** problem.
- If we choose $0 < \gamma < 1$, the infinite sum converges, even when $n \rightarrow \infty$.
- Infinite horizon problems lend themselves to analysis because terms are weighted by $\gamma^{i-1}$, there is often little practical difference between infinite and finite horizon solutions.

# Expected Utility

- If we want to plan into the future, we need to deal with the fact that the future is uncertain.
- We can do this by computing an ***expected utility***.
- The expectation is taken with respect to the unknown future states (i.e., $X_2 \dots X_{n+1}$ are random).

$$\mathrm{E}\left[U\left(a_{1,} \dots, a_n, x_1, X_2, \dots X_{n+1}\right)\right] = \mathrm{E}[R(x_1, a_1, X_2) + \gamma R(X_2, a_2, X_3) + \cdots \gamma^{n-1} R(X_n, a_n, X_{n+1})\ ]$$

- The expected utility is a function of the action sequence $a_1 \dots a_n$ and the current state $x_1$:

$$\bar{U}(a_1 \dots a_n, x_1) = \mathrm{E}\left[U\left(a_{1,} \dots, a_n, x_1, X_2, \dots X_{n+1}\right)\right]$$

- A simple (brute force) planning algorithm:

$$a_1^* \dots a_n^* = \arg \max_{a_1 \dots a_n} \bar{U}(a_1 \dots a_n, x_1)$$

***This is an open-loop approach! There is no feedback.***

## Expected Utility

- For a finite horizon problem, we can compute the expectation using brute force:

$$\bar{U}(a_1 \dots a_n, x_1) = \sum_{(x_2 \dots x_{n+1}) \in X^n} P(x_2, \dots x_{n+1}) \sum_{i=1}^{n} \gamma^{i-1} R(x_i, a_i, x_{i+1})$$

- For every possible state sequence:
  - Compute the probability of that sequence (given the action sequence).
  - Compute the utility for that sequence.
  - Weight the utility of the sequence by the probability of achieving that sequence.
- This approach is exponential in the horizon length, so we can't use it for large, or even moderate, values of $n$. ☹

# Control Tape Rollouts

- We can use sampling to approximate the expected utility.
- Call the action sequence $a_1 \dots a_n$ a control tape.
- Generate $L$ sample state trajectories: $x_1^l, x_2^l, \dots x_{n+1}^l$ for $1 < l \leq L$
- Compute the sum:

$$\tilde{U}(a_1 \dots a_n, x_1) = \frac{1}{L}\sum_{l=1}^{L} \sum_{i=1}^{n} \gamma^{i-1} R(x_i^l, a_i, x_{i+1}^l)$$

- Remember the weak law of large numbers: As $L$ becomes large,

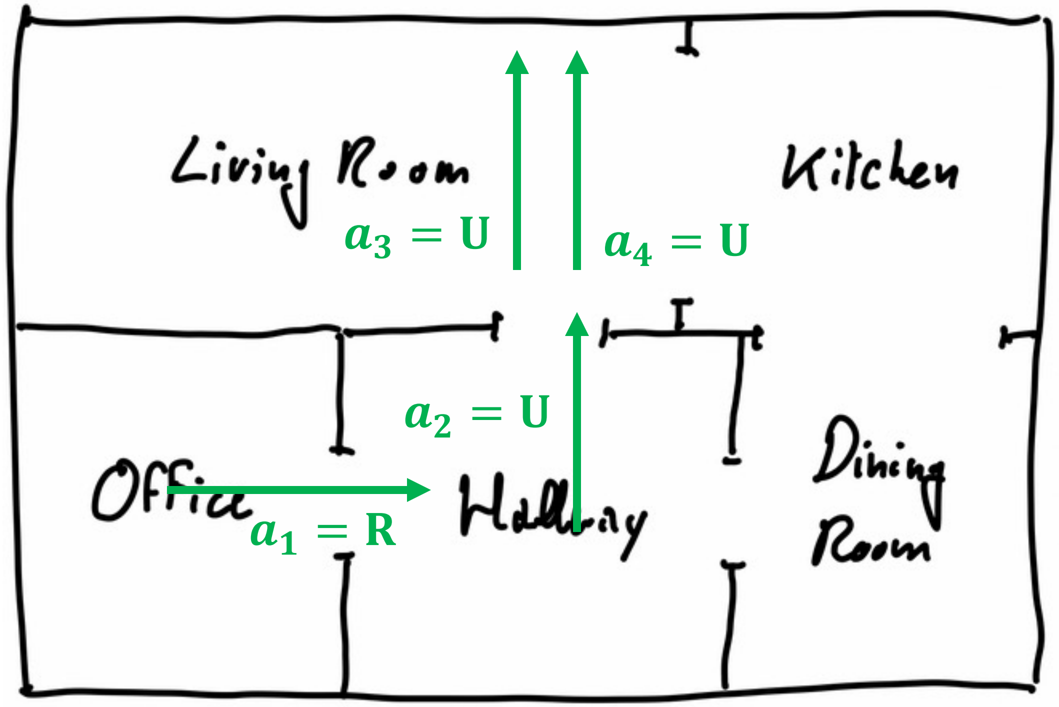
$$\tilde{U}(a_1 \dots a_n, x_1) \rightarrow E[U(a_1 \dots a_n, x_1)]$$

# Policies

Suppose we start in the office.
What would be the best choice for the next four actions?

Maybe R, U, U, U?

This seems perfect!
1. The first action takes us to the hallway.
2. The second action takes us to the Living Room
3. The third and fourth actions keep us in the Living Room.



# Policies

Suppose we start in the office.
What would be the best choice for the next four actions?

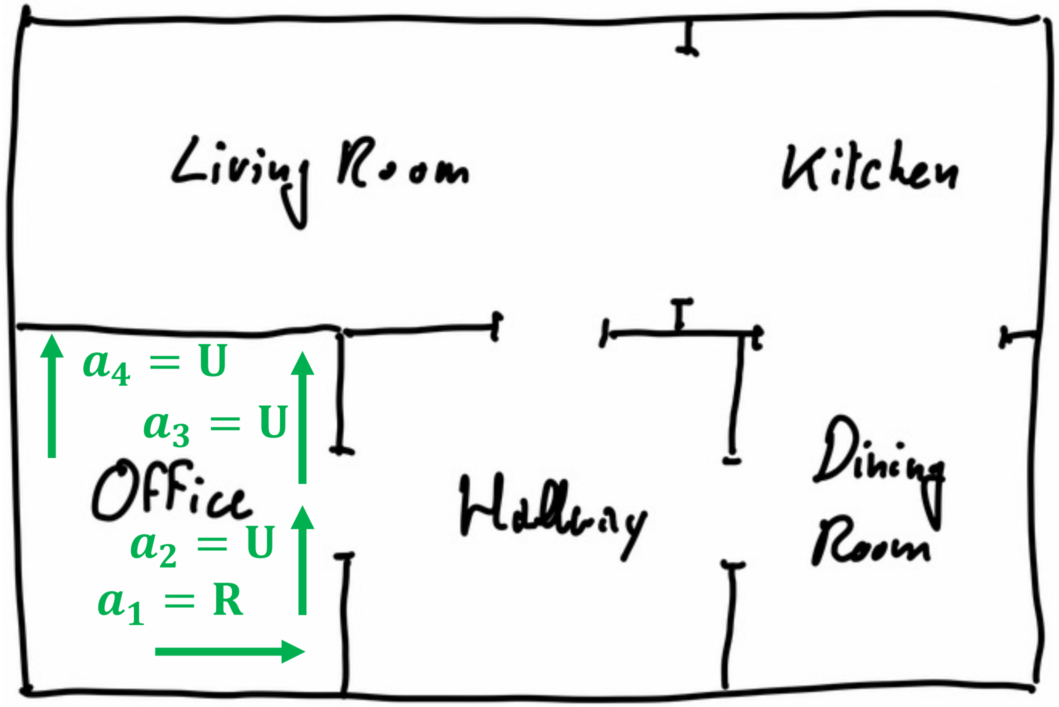
Maybe R, U, U, U?

But what if the first action fails to get us to the Hallway?

1. The first action stays in the Office.
2. The next three actions keep us in the Office.

This sequence of actions is great if the first action succeeds, but horrible if the first action fails.

➢ ***It would be nice if the robot could make decisions based on the current state, instead of preplanning the entire action sequence!***

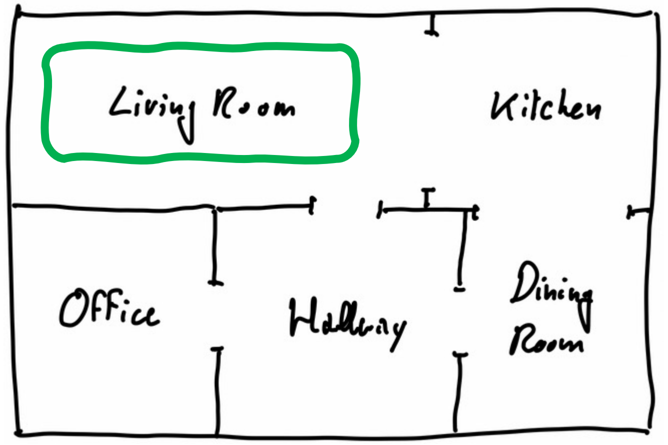


# Policies $\pi: \mathcal{X} \rightarrow \mathcal{A}$

- Because actions are non-deterministic, fixed plans are brittle and prone to failure.
- Better to have a *state-dependent* plan!
- A policy $\pi(X)$ is a function that specifies which action to take in each state.
- Let us come up with a policy together:

- $\pi(L) = U$ (Stay in the Living Room)
- $\pi(K) = L$ (Try for the Living Room)
- $\pi(O) = R$ (Try for the Hallway)
- $\pi(H) = U$ (Try for the Living Room)
- $\pi(D) = U$ (Try for the Kitchen)

➢ ***This is nice for very simple cases, but we need something more general…***



# The Value Function for a Policy

Recall the Expected Utility

$$\bar{U}(a_1 \dots a_n, x_1) = \mathrm{E}[R(x_1, a_1, X_2) + \gamma R(X_2, a_2, X_3) + \cdots \gamma^{n-1} R(X_n, a_n, X_{n+1})\ ]$$

- For a policy, we can define this similarly:

$$\bar{U}(\pi, n, x_1) \doteq E\ [R(x_1, \pi(x_1), X_2) + \gamma\ R(X_2, \pi(X_2), X_3) + \cdots + \gamma^{n-1} R(X_n, \pi(X_n), X_{n+1})]$$

- Can be extended to infinite horizon policy, defining the value function:

$$V^{\pi}(x_1) \doteq E\ [R(x_1, \pi(x_1), X_2) + \gamma\ R(X_2, \pi(X_2), X_3) + \gamma^2 R(X_3, \pi(X_3), X_4) + \cdots]$$

➢Let’s take a closer look at this…

# The Value Function for a Policy

- The Value Function for a policy $\pi$ is the expected value of the infinite horizon utility:

$$V^{\pi}(x_1) \doteq E\left[\sum_{i=1}^{\infty} \gamma^{i-1} R(X_i, a_i, X_{i+1})\right]$$

- Note that the expectation is taken with respect to the random states $X_2, X_3$ , ...

- Of course, above holds for arbitrary $x_t$, not just $x_1$

$$V^{\pi}(x_t) \doteq E\left[\sum_{i=t}^{\infty} \gamma^{i-t} R(X_i, a_i, X_{i+1})\right]$$

# Recursive Definition of $V^{\pi}$

$$V^{\pi}(x_1) = E[R(x_1, \pi(x_1), X_2) + \gamma R(X_2, \pi(X_2), X_3) + \gamma^2 R(X_3, \pi(X_3), X_4) + \ \ldots]$$

- This is a fairly complicated expectation, taken with respect to the infinite sequence of random variables $X_2$, $X_3$ ...
- Let's break it down into two pieces:
    - Expectation with respect to $X_2$
    - Expectation with respect to $X_3$, $X_4$ ...
    - To do this, write out the actual definition of expectation w.r.t. $X_2$

# Recursive Definition of $V^{\pi}$

$$V^{\pi}(x_1) = E[R(x_1, \pi(x_1), X_2) + \gamma R(X_2, \pi(X_2), X_3) + \gamma^2 R(X_3, \pi(X_3), X_4) + \ \dots]$$

$$V^{\pi}(x_1) = \sum_{x_2} P(x_2 | x_1, \pi(x_1))\{R(x_1, \pi(x_1), x_2) + \gamma E[R(x_2, \pi(x_2), X_3) + \gamma R(X_3, \pi(X_3), X_4) + \ \dots]\}$$

Expectation with respect to $X_2$

Expectation with respect to $X_3$, $X_4$ …

But Notice: The expectation in red is exactly $V^{\pi}(x_2)$!

# Recursive Definition of $V^{\pi}$

$$V^{\pi}(x_1) = E[R(x_1, \pi(x_1), X_2) + \gamma R(X_2, \pi(X_2), X_3) + \gamma^2 R(X_3, \pi(X_3), X_4) + \ \dots]$$

$$V^{\pi}(x_1) = \sum_{x_2} P(x_2 | x_1, \pi(x_1)) \{R(x_1, \pi(x_1), x_2) + \gamma E[R(x_2, \pi(x_2), X_3) + \gamma R(X_3, \pi(X_3), X_4) + \ \dots]\}$$

$$V^{\pi}(x_1) = \sum_{x_2} P(x_2 | x_1, \pi(x_1)) \{R(x_1, \pi(x_1), x_2) + \gamma V^{\pi}(x_2)\}$$

Recall: Expectation is linear.

➢ Let’s take the summation, and distribute it over the two terms in brackets.

# Recursive Definition of $V^{\pi}$

$$V^{\pi}(x_1) = E[R(x_1, \pi(x_1), X_2) + \gamma R(X_2, \pi(X_2), X_3) + \gamma^2 R(X_3, \pi(X_3), X_4) + \ \ldots]$$

$$V^{\pi}(x_1) = \sum_{x_2} P(x_2 | x_1, \pi(x_1)) \{R(x_1, \pi(x_1), x_2) + \gamma E[R(x_2, \pi(x_2), X_3) + \gamma R(X_3, \pi(X_3), X_4) + \ \ldots]\}$$

$$V^{\pi}(x_1) = \sum_{x_2} P(x_2 | x_1, \pi(x_1)) \{R(x_1, \pi(x_1), x_2) + \gamma V^{\pi}(x_2)\}$$

$$V^{\pi}(x_1) \boxed{= \sum_{x_2} P(x_2 | x_1, \pi(x_1)) R(x_1, \pi(x_1), x_2)} + \gamma \sum_{x_2} P(x_2 | x_1, \pi(x_1)) V^{\pi}(x_2)$$

This term is our old friend, the expected reward for applying $\pi(x_1)$ in state $x_1$ : $\bar{R}(x_1, a)$

# Recursive Definition of $V^\pi$

$$V^\pi(x_1) = E[R(x_1, \pi(x_1), X_2) + \gamma R(X_2, \pi(X_2), X_3) + \gamma^2 R(X_3, \pi(X_3), X_4) + \ \dots]$$

$$V^\pi(x_1) = \sum_{x_2} P(x_2|x_1, \pi(x_1))\{R(x_1, \pi(x_1), x_2) + \gamma E[R(x_2, \pi(x_2), X_3) + \gamma R(X_3, \pi(X_3), X_4) + \ \dots]\}$$

$$V^\pi(x_1) = \sum_{x_2} P(x_2|x_1, \pi(x_1))\{R(x_1, \pi(x_1), x_2) + \gamma V^\pi(x_2)\}$$

$$V^\pi(x_1) = \sum_{x_2} P(x_2|x_1, \pi(x_1))R(x_1, \pi(x_1), x_2) \ + \gamma \sum_{x_2} P(x_2|x_1, \pi(x_1))V^\pi(x_2)$$

$$\boxed{V^\pi(x) = \bar{R}(x, \pi(x)) \ + \gamma \sum_{x'} P(x'|x, \pi(x))V^\pi(x')}$$

## Recursive Definition of $V^{\pi}$

The value function for policy $\pi$ can be written recursively as

$$V^{\pi}(x) = \bar{R}(x, \pi(x)) + \gamma \sum_{x'} P(x'|x, \pi(x)) V^{\pi}(x')$$

***Expected reward for the next step***

***Discounted expectation of value function from the next step***

Note that the sum is taken over all possible next states $x'$

# Approximating $V^{\pi}$

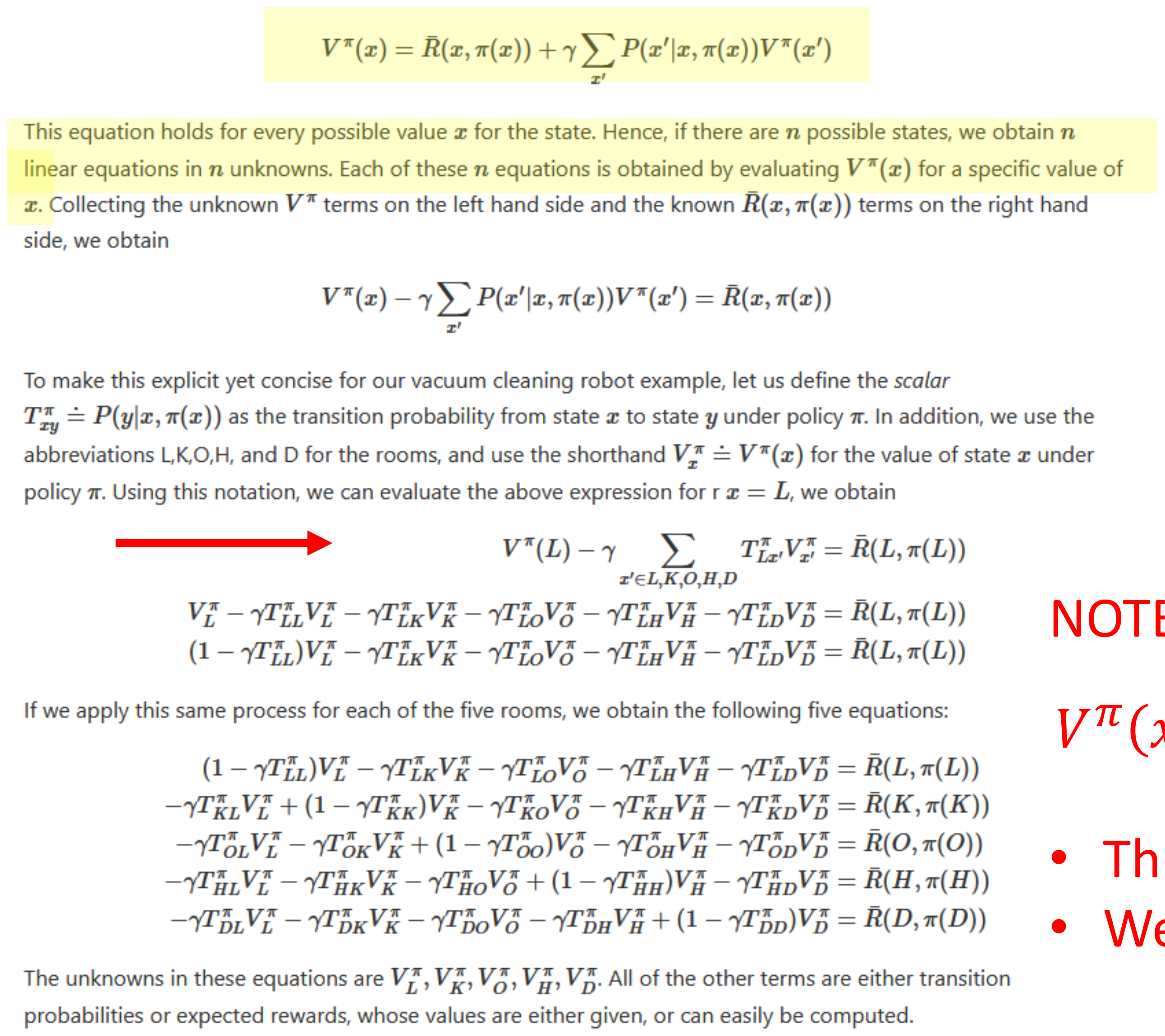
- Just as with expected utility, we can use sampling to approximate the value function.
- Generate $L$ sample trajectories: $x_1^l, x_2^l, \dots x_{n+1}^l$ for $1 < l \leq L$

$$\bar{V}^{\pi}(x_1) \approx \frac{1}{N}\sum_{l=1}^{N}\sum_{k=1}^{n}\gamma^{k-1}R(x_k^l, \pi\left(x_k^l\right), x_{k+1}^l)$$

- Remember the weak law of large numbers: As $L$ becomes large,

$$\bar{V}^{\pi}(x_1) \rightarrow E\left[\sum_{i=1}^{\infty}\gamma^{i-1}R(X_i, a_i, X_{i+1})\right]$$

# Exact Computation for $V^\pi$



$$V^\pi(x) = \bar{R}(x, \pi(x)) + \gamma \sum_{x'} P(x'|x, \pi(x)) V^\pi(x')$$

This equation holds for every possible value $x$ for the state. Hence, if there are $n$ possible states, we obtain $n$ linear equations in $n$ unknowns. Each of these $n$ equations is obtained by evaluating $V^\pi(x)$ for a specific value of $x$. Collecting the unknown $V^\pi$ terms on the left hand side and the known $\bar{R}(x, \pi(x))$ terms on the right hand side, we obtain

$$V^\pi(x) - \gamma \sum_{x'} P(x'|x, \pi(x)) V^\pi(x') = \bar{R}(x, \pi(x))$$

To make this explicit yet concise for our vacuum cleaning robot example, let us define the *scalar* $T^\pi_{xy} \doteq P(y|x, \pi(x))$ as the transition probability from state $x$ to state $y$ under policy $\pi$. In addition, we use the abbreviations L,K,O,H, and D for the rooms, and use the shorthand $V^\pi_x \doteq V^\pi(x)$ for the value of state $x$ under policy $\pi$. Using this notation, we can evaluate the above expression for r $x = L$, we obtain

$$\longrightarrow \quad V^\pi(L) - \gamma \sum_{x' \in L,K,O,H,D} T^\pi_{Lx'} V^\pi_{x'} = \bar{R}(L, \pi(L))$$
$$V^\pi_L - \gamma T^\pi_{LL} V^\pi_L - \gamma T^\pi_{LK} V^\pi_K - \gamma T^\pi_{LO} V^\pi_O - \gamma T^\pi_{LH} V^\pi_H - \gamma T^\pi_{LD} V^\pi_D = \bar{R}(L, \pi(L))$$
$$(1 - \gamma T^\pi_{LL}) V^\pi_L - \gamma T^\pi_{LK} V^\pi_K - \gamma T^\pi_{LO} V^\pi_O - \gamma T^\pi_{LH} V^\pi_H - \gamma T^\pi_{LD} V^\pi_D = \bar{R}(L, \pi(L))$$

If we apply this same process for each of the five rooms, we obtain the following five equations:

$$(1 - \gamma T^\pi_{LL}) V^\pi_L - \gamma T^\pi_{LK} V^\pi_K - \gamma T^\pi_{LO} V^\pi_O - \gamma T^\pi_{LH} V^\pi_H - \gamma T^\pi_{LD} V^\pi_D = \bar{R}(L, \pi(L))$$
$$-\gamma T^\pi_{KL} V^\pi_L + (1 - \gamma T^\pi_{KK}) V^\pi_K - \gamma T^\pi_{KO} V^\pi_O - \gamma T^\pi_{KH} V^\pi_H - \gamma T^\pi_{KD} V^\pi_D = \bar{R}(K, \pi(K))$$
$$-\gamma T^\pi_{OL} V^\pi_L - \gamma T^\pi_{OK} V^\pi_K + (1 - \gamma T^\pi_{OO}) V^\pi_O - \gamma T^\pi_{OH} V^\pi_H - \gamma T^\pi_{OD} V^\pi_D = \bar{R}(O, \pi(O))$$
$$-\gamma T^\pi_{HL} V^\pi_L - \gamma T^\pi_{HK} V^\pi_K - \gamma T^\pi_{HO} V^\pi_O + (1 - \gamma T^\pi_{HH}) V^\pi_H - \gamma T^\pi_{HD} V^\pi_D = \bar{R}(H, \pi(H))$$
$$-\gamma T^\pi_{DL} V^\pi_L - \gamma T^\pi_{DK} V^\pi_K - \gamma T^\pi_{DO} V^\pi_O - \gamma T^\pi_{DH} V^\pi_H + (1 - \gamma T^\pi_{DD}) V^\pi_D = \bar{R}(D, \pi(D))$$

The unknowns in these equations are $V^\pi_L, V^\pi_K, V^\pi_O, V^\pi_H, V^\pi_D$. All of the other terms are either transition probabilities or expected rewards, whose values are either given, or can easily be computed.

$$T^\pi_{xy} \triangleq P(X' = y | X = x, \pi(x))$$

$$V^\pi_x \triangleq V^\pi(x)$$

NOTE:

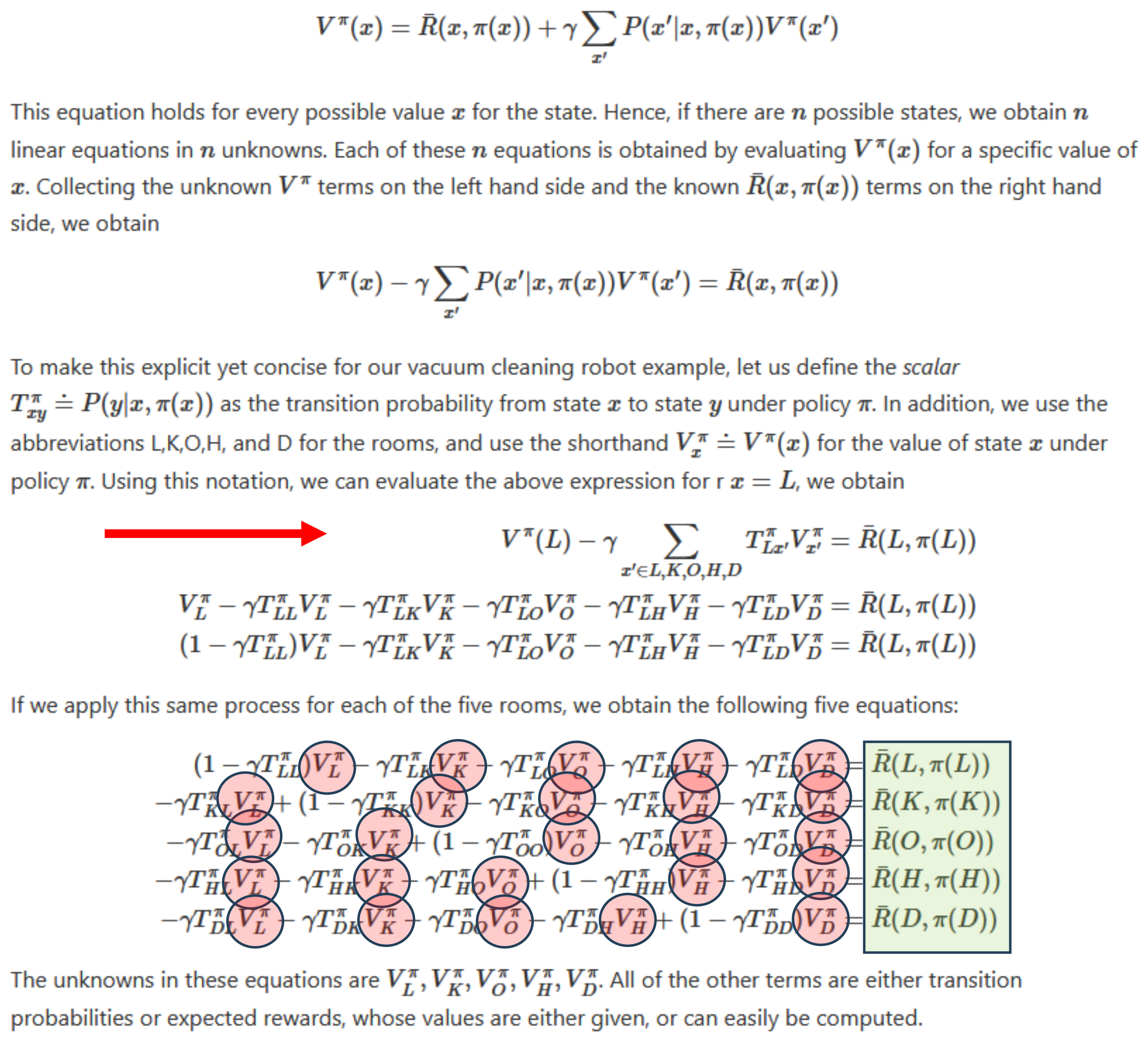
$$V^\pi(x) = \bar{R}(x, \pi(x)) + \gamma \sum_{x'} P(x'|x, \pi(x)) V^\pi(x')$$

- This is a linear equation.
- We can solve it using linear algebra!

# Exact Computation for $V^{\pi}$

$$T_{xy}^{\pi} \triangleq P(X' = y | X = x, \pi(x))$$

$$V_x^{\pi} \triangleq V^{\pi}(x)$$



$$V^{\pi}(x) = \bar{R}(x,\pi(x)) + \gamma \sum_{x'} P(x'|x,\pi(x)) V^{\pi}(x')$$

This equation holds for every possible value $x$ for the state. Hence, if there are $n$ possible states, we obtain $n$ linear equations in $n$ unknowns. Each of these $n$ equations is obtained by evaluating $V^{\pi}(x)$ for a specific value of $x$. Collecting the unknown $V^{\pi}$ terms on the left hand side and the known $\bar{R}(x,\pi(x))$ terms on the right hand side, we obtain

$$V^{\pi}(x) - \gamma \sum_{x'} P(x'|x,\pi(x)) V^{\pi}(x') = \bar{R}(x,\pi(x))$$

To make this explicit yet concise for our vacuum cleaning robot example, let us define the *scalar* $T_{xy}^{\pi} \doteq P(y|x,\pi(x))$ as the transition probability from state $x$ to state $y$ under policy $\pi$. In addition, we use the abbreviations L,K,O,H, and D for the rooms, and use the shorthand $V_x^{\pi} \doteq V^{\pi}(x)$ for the value of state $x$ under policy $\pi$. Using this notation, we can evaluate the above expression for r $x = L$, we obtain

$$V^{\pi}(L) - \gamma \sum_{x' \in L,K,O,H,D} T_{Lx'}^{\pi} V_{x'}^{\pi} = \bar{R}(L,\pi(L))$$
$$V_L^{\pi} - \gamma T_{LL}^{\pi} V_L^{\pi} - \gamma T_{LK}^{\pi} V_K^{\pi} - \gamma T_{LO}^{\pi} V_O^{\pi} - \gamma T_{LH}^{\pi} V_H^{\pi} - \gamma T_{LD}^{\pi} V_D^{\pi} = \bar{R}(L,\pi(L))$$
$$(1 - \gamma T_{LL}^{\pi}) V_L^{\pi} - \gamma T_{LK}^{\pi} V_K^{\pi} - \gamma T_{LO}^{\pi} V_O^{\pi} - \gamma T_{LH}^{\pi} V_H^{\pi} - \gamma T_{LD}^{\pi} V_D^{\pi} = \bar{R}(L,\pi(L))$$

If we apply this same process for each of the five rooms, we obtain the following five equations:

$$\begin{aligned}
(1 - \gamma T_{LL}^{\pi}) V_L^{\pi} - \gamma T_{LK}^{\pi} V_K^{\pi} - \gamma T_{LO}^{\pi} V_O^{\pi} - \gamma T_{LH}^{\pi} V_H^{\pi} - \gamma T_{LD}^{\pi} V_D^{\pi} &= \bar{R}(L,\pi(L)) \\
-\gamma T_{KL}^{\pi} V_L^{\pi} + (1 - \gamma T_{KK}^{\pi}) V_K^{\pi} - \gamma T_{KO}^{\pi} V_O^{\pi} - \gamma T_{KH}^{\pi} V_H^{\pi} - \gamma T_{KD}^{\pi} V_D^{\pi} &= \bar{R}(K,\pi(K)) \\
-\gamma T_{OL}^{\pi} V_L^{\pi} - \gamma T_{OK}^{\pi} V_K^{\pi} + (1 - \gamma T_{OO}^{\pi}) V_O^{\pi} - \gamma T_{OH}^{\pi} V_H^{\pi} - \gamma T_{OD}^{\pi} V_D^{\pi} &= \bar{R}(O,\pi(O)) \\
-\gamma T_{HL}^{\pi} V_L^{\pi} - \gamma T_{HK}^{\pi} V_K^{\pi} - \gamma T_{HO}^{\pi} V_O^{\pi} + (1 - \gamma T_{HH}^{\pi}) V_H^{\pi} - \gamma T_{HD}^{\pi} V_D^{\pi} &= \bar{R}(H,\pi(H)) \\
-\gamma T_{DL}^{\pi} V_L^{\pi} - \gamma T_{DK}^{\pi} V_K^{\pi} - \gamma T_{DO}^{\pi} V_O^{\pi} - \gamma T_{DH}^{\pi} V_H^{\pi} + (1 - \gamma T_{DD}^{\pi}) V_D^{\pi} &= \bar{R}(D,\pi(D))
\end{aligned}$$

The unknowns in these equations are $V_L^{\pi}, V_K^{\pi}, V_O^{\pi}, V_H^{\pi}, V_D^{\pi}$. All of the other terms are either transition probabilities or expected rewards, whose values are either given, or can easily be computed.

$$\begin{bmatrix} (1-\gamma)T_{LL}^{\pi} & \cdots & -\gamma T_{LD}^{\pi} \\ \vdots & \ddots & \vdots \\ -\gamma T_{DL}^{\pi} & \cdots & (1-\gamma)T_{DD}^{\pi} \end{bmatrix} \begin{bmatrix} V_L^{\pi} \\ V_K^{\pi} \\ V_O^{\pi} \\ V_H^{\pi} \\ V_D^{\pi} \end{bmatrix} = \begin{bmatrix} \bar{R}(L,\pi(L)) \\ \bar{R}(K,\pi(K)) \\ \bar{R}(O,\pi(O)) \\ \bar{R}(H,\pi(H)) \\ \bar{R}(D,\pi(D)) \end{bmatrix}$$

## Optimal Value Function

The Optimal Value Function is merely the max over all possible policies of $V^{\pi}$:

$$V^*(x) = \max_{\pi} V^{\pi}(x)$$

$$= \max_{\pi} \left\{ \bar{R}(x, \pi(x)) \ + \gamma \sum_{x'} P(x'|x, \pi(x)) V^{\pi}(x') \right\}$$

$$= \max_{a} \left\{ \bar{R}(x, a) \ + \gamma \sum_{x'} P(x'|x, a)) V^{*}(x') \right\}$$

# The Bellman Equation

***<u>Principle of Optimality</u>***: *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.*

$$V^*(x) = \max_{\pi}\left\{\bar{R}(x,\pi(x)) + \gamma \sum_{x'} P(x'|x,\pi(x))V^{\pi}(x')\right\}$$

$\pi(x) = a$ $\quad\quad$ $\pi(x) = a$ $\quad\quad$ ***Principle of Optimality***

$$= \max_{a}\left\{\bar{R}(x,a) + \gamma \sum_{x'} P(x'|x,a)V^*(x')\right\}$$

R. Bellman and R. Kalaba, "Dynamic programming and adaptive processes: Mathematical foundation," in IRE Transactions on Automatic Control, vol. AC-5, no. 1, pp. 5-10, Jan. 1960, doi: 10.1109/TAC.1960.6429288.

# The Bellman Equation

***Another look***:

$$V^*(x) = \max_a \left\{ \bar{R}(x,a) + \gamma \sum_{x'} P(x'|x,a) V^*(x') \right\}$$

Sub-solution in the recursion

- $V^*(x)$ is on both sides of the equal sign → recursive definition!
- This is ***not*** a linear equation, because ***max is not a linear operation***!

# Optimal Policy

Given the $\boldsymbol{V^*(x)}$, computing the optimal policy is a straightforward optimization:

$$\boldsymbol{\pi^*(x) = arg\,\underset{a}{max}\left\{\bar{R}(x,a)\ + \gamma \sum_{x'} P(x'|x,a))V^*(x')\right\}}$$

For convenience, we define the $\boldsymbol{Q^*}$ function as

$$\boldsymbol{Q^*(x,a) = \bar{R}(x,a)\ + \gamma \sum_{x'} P(x'|x,a))V^*(x')}$$

and we can write the optimal policy as:

$$\boldsymbol{\pi^*(x) = arg\,\underset{a}{max}\,Q^*(x,a)}$$

# Policy Iteration

Start with a random policy $\pi^0$, and repeat until convergence:

1. Compute the value function $V^{\pi^k}$
2. Improve the policy for each state $x$ using the update rule:

$$\boldsymbol{\pi}^{k+1}(\boldsymbol{x}) \leftarrow \underset{\boldsymbol{a}}{\boldsymbol{arg\,max}} \left\{ \bar{\boldsymbol{R}}(\boldsymbol{x}, \boldsymbol{a}) + \boldsymbol{\gamma} \sum_{\boldsymbol{x}'} \boldsymbol{P}(\boldsymbol{x}' | \boldsymbol{x}, \boldsymbol{a}) \boldsymbol{V}^{\boldsymbol{\pi}^{k}}(\boldsymbol{x}') \right\}$$

Improve the
current guess for $\boldsymbol{\pi}^*$

The current guess for $\boldsymbol{\pi}^*$

## Value Iteration

Start with a random value function $V^0$, and repeat until convergence:

- Improve the value function $V^k$ using the update rule:

$$\boldsymbol{V^{k+1}(x) \leftarrow \max_a \left\{ \bar{R}(x,a) + \gamma \sum_{x'} P(x'|x,a)) V^k(x') \right\}}$$

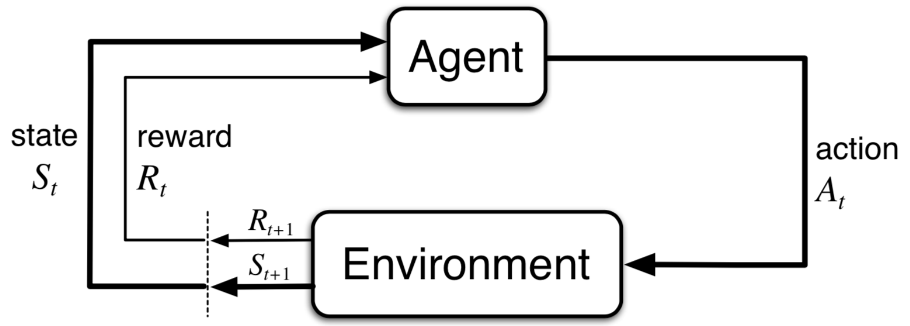
Improve the current guess for $\boldsymbol{V}^*$

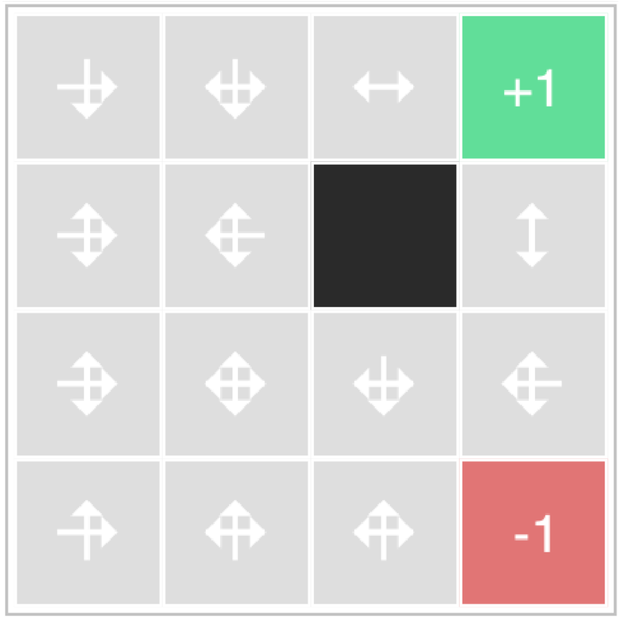
The current guess for $\boldsymbol{V}^*$

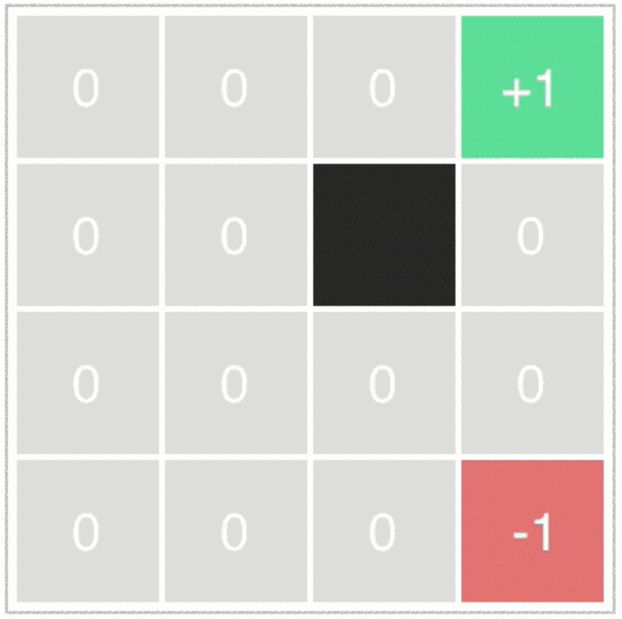
***Note:***

- ***For policy iteration we used arg max and selected an action, which implicitly updates the value function.***
- ***For value iteration, we use max, and update directly the value function.***

# Value Interaction in action

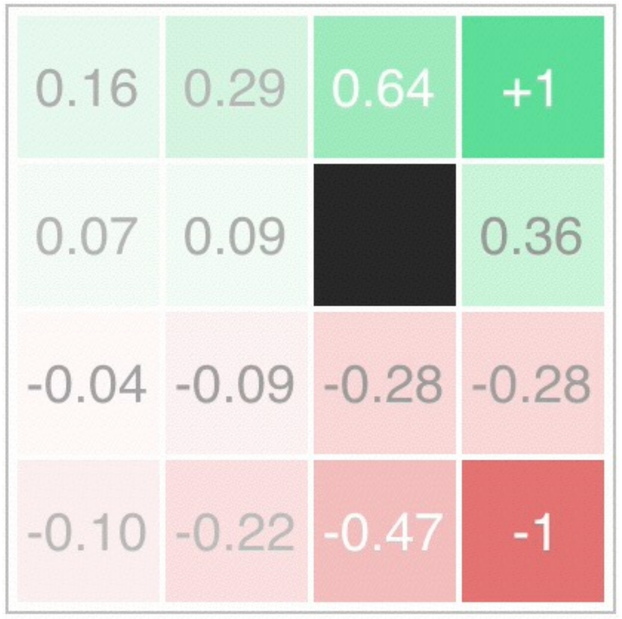






| | | | |
|---|---|---|---|
| 0 | 0 | 0 | +1 |
| 0 | 0 | | 0 |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | -1 |

Iteration 1



| | | | |
|---|---|---|---|
| 0.16 | 0.29 | 0.64 | +1 |
| 0.07 | 0.09 | | 0.36 |
| -0.04 | -0.09 | -0.28 | -0.28 |
| -0.10 | -0.22 | -0.47 | -1 |

Final Values