# Introduction, Markov, Bellman Equation

Wen Yu
Departamento de Control Automático
CINVESTAV-IPN

# Learning algorithms

Fundamental challenge in AI and machine learning is:
Learn to make good decisions under uncertainty

1. Supervised learning: regression, classification, modeling (Teaching)
2. Unsupervised learning: clustering, dimension reduction
3. Reinforcement Learning: learn what to do; how to map situations to actions by maximizing a numerical reward signal. (Punishment)

# Application of RL

- Video game
- Robotics
- Education
- Health care
- Solve optimization problems

# Fundamental problem: Differences with the other machining learnings

- Optimization
- Delayed consequence
- Exploration: experiences
- Generalization: *Policy is mapping from past experience to action, why not just pre-program a policy? We never see before*

# History of reinforcement learning

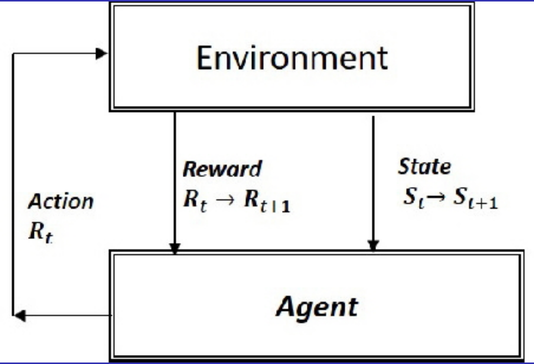
There are two main roads (1980s)

1. Concerns learning by trial and error, originated in the psychology of animal learning
2. Concerns the problem of optimal control, did not involve learning, dynamic programming

Concerning temporal-difference methods
1989, The temporal-difference and optimal control were fully brought together with: Q-learning

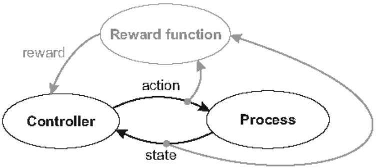
# Reinforcement Learning vs other learning methods

- Reinforcement learning: uses training information to evaluate the actions
- Other learning methods: use training information to give correct actions
- Reinforcement learning: active exploration for searching good action
- Other learning methods: the action is the best or the worst




Learning problems facing a decision-maker interacting with its environment
Goal: Select actions to maximize total expected future reward
immediate/long term reward
strategic behaviors to obtain high reward

# Reinforcement Learning/dynamic programming vs control




In dynamic programming (DP) and reinforcement learning (RL), a controller (agent, decision maker) interacts with a process (environment), by means of three signals:

1. state signal, which describes the state of the process
2. action signal, which allows the controller to influence the process
3. reward signal, which provides the controller with feedback on its immediate performance.

# References


- R.S. Sutton, A.G. Barto, *Reinforcement Learning: An Introduction*, 2nd edition, The MIT Press, 2018
- L.Busoniu, R.Babuska, Bart.Schutter, D.Ernst, R*einforcement Learning and Dynamic Programming Using Function Approximators Technology*, CRC Press, 2010

# Notation in RL/DP vs control

RL vs DP

- RL uses Max/Value, DP uses Min/Cost
- State value $\longrightarrow$ State cost.
- Value (or state-value) function $\longrightarrow$ Cost function.

Controller

- Agent $\longrightarrow$ controller
- Action $\longrightarrow$ control
- Environment $\longrightarrow$ Dynamic system

# Probability Theory

The probability of the variable $A$ takes $a$ is

$$0 \leq P\left(A = a\right) \leq 1$$

Alternatives ->"add"

$$P\left(A = a_1 \text{ or } A = a_2\right) = P\left(A = a_1\right) + P\left(A = a_2\right)$$

Normalisation

$$\sum_{\text{all possible } a} P\left(A = a\right) = 1$$

Joint probability

$$P\left(A = a, B = b\right) \text{ the probability that both } A = a \text{ and } B = b \text{ occur}$$

Marginalisation

$$P\left(A = a\right) = \sum_{\text{all possible } b} P\left(A = a, B = b\right)$$

# Probability Theory

ConditionalProbability

$P(A = a \mid B = b)$ the probability that $A = a$ occurs given the knowledge $B$

ProductRule

$$\begin{aligned} & P(A = a, B = b) \\ = & P(A = a) P(B = b \mid A = a) \\ = & P(B = b) P(A = a \mid B = b) \end{aligned}$$

Independence, iff $A$ and $B$ are independent:

$$\begin{aligned} P(A = a \mid B = b) &= P(A = a) \\ P(B = b \mid A = a) &= P(B = b) \\ P(A = a, B = b) &= P(A = a) P(B = b) \end{aligned}$$

Bayes Rule

$$P(A = a \mid B = b) = \frac{P(B=b|A=a)P(A=a)}{P(B=b)}$$
$$P(A \mid B) = \frac{P(B|A)P(A)}{P(B)}$$

Continuous variables, "Sum" $\rightarrow$ "Integral", the probability that $X$ lies between $x$ and $(x + dx)$ is $p(x)dx$, $p(x)$ is a probability density function

$$P(x_1 \leq X \leq x_2) = \int_{x_1}^{x_2} p(x)dx$$
$$\int_{-\infty}^{+\infty} p(x)dx = 1$$
$$\int_{-\infty}^{+\infty} p(x, y)dy = p(x)$$

# Probability Theory

Expectations: Averages over a time series $X$

$$E[A] = \sum_a P(A = a) A$$
$$E[X] = \int_{-\infty}^{+\infty} p(x) x dx$$

If $X$ and $Y$ are independent

$$E[X \times Y] = E[X] \times E[Y]$$

# Series Theory

Geometric series

$$1 + \alpha + a^2 + \cdots = \sum_{i=1}^{n} a^i = s_n$$
$$\lim_{n\to\infty} = \frac{1}{1-\alpha}$$

Moving average

$$\bar{X} = \frac{1}{m}\sum_{i=n-m+1}^{n} x_i$$

Weighted moving average

$$\bar{X} = \frac{1}{\sum_{i=n-m+1}^{n} \alpha_i}\sum_{i=n-m+1}^{n} \alpha_i x_i = \frac{1}{\sum_{i=1}^{m} \alpha_i}\sum_{i=1}^{m} \alpha_i x_i$$

Exponentially weighted average

$$\bar{X} = \frac{1}{\sum_{i=1}^{\infty} \alpha_i}\sum_{i=1}^{\infty} \alpha^i x_i = (1-\alpha)\sum_{i=1}^{\infty} \alpha^i x_i$$

# Markov Processes (MP)

Sequence of past observations, actions and rewards

$$h_t = \{a_1, o_1, r_1, \cdots a_t, o_t, r_t\}$$

Markov assumption:

- Each event depends only on in the previous event, this process requires the nature of "memoryless"
- Or in order to predict the future, you only need to know the current state of the environment

The $s_t$ is Markov iff

$$P\{s_{t+1} \mid s_t, a_t\} = P\{s_{t+1} \mid h_t, a_t\}$$

Prior examples $h_t$

# No Markov Processes

The current blood pressure is the state, the action is whether to take medication or not, it is Not Markov, because there are many other states to influence the action and also all history blood pressures.
To be Markov

$$s_t = h_t$$

In practice: recent observation

$$s_t = o_t$$

# Notation

MP is the sets of states and actions

$$MDP = \{\mathcal{S}, \mathcal{A}, \mathcal{P}\}$$

- $\mathcal{P}$ is random variable defined discrete probability distributions of $\mathcal{S}$ and $\mathcal{A}$
- $s$ and $\bar{s}$ are values of the random variables $S_{t-1}$ and $S_t$, $s$,$\bar{s} \in \mathcal{S}$
- $a$ is the value of the action $A_t$, $a \in \mathcal{A}(s)$

The process of $S_{t-1} - A_{t-1} \rightarrow S_t - A_t - R_t$ is

$$p\left\{S_t = \bar{s}, R_t = r \mid S_{t-1} = s, A_{t-1} = a\right\}$$
$$or \quad p\left(\bar{s}, r \mid s, a\right)$$

where $p(\cdot)$ is the dynamics of the MP, $p$ specifies a probability distribution for each choice of $s$ and $a$, so

$$\sum_{\bar{s} \in \mathcal{S}} \sum_{r \in \mathcal{R}} p\left(\bar{s}, r \mid s, a\right) = 1$$

# Components of RL

- Model: how world changes in response to agent's action
- Policy: mapping agent's states to action
- Value function: rewards from state and action

1) Transition or dynamic model, to predict next agent state

$$p\left(s_{t+1} = s' \mid s_t = s, a_t = a\right)$$

2) Reward model

$$r\left(s_t = s, a_t = a\right) = \mathbb{E}\left[\left(r_t \mid s_t = s, a_t = a\right)\right]$$

# Reward, Return and Value

Horizon: number of time step in each episode, it can be infinite or finite

Return $G$: Discounted sum of rewards from $t$ to horizon is

$$G_t = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \cdots$$

Value $V$: sum of future reward $r$ under a particular policy $\pi$

$$\begin{gathered} V^{\pi}\left(s_t = s\right) = \mathbb{E}\left\{G_t \mid s_t = s\right\} \\ = \mathbb{E}_{\pi}\left[r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \cdots \mid s_t = s\right] \end{gathered}$$

where $0 \leq \gamma \leq 1$ is discount factor

# Infinite horizon and continuing task

$T = \infty$

$$G_\infty = \sum_{i=0}^{\infty} \gamma^k R_{t+1+i}$$

Although the sum of an infinite number of terms, it is still finite if the reward is nonzero and constant, and $\gamma < 1$.Foe example $R_t = 1$

$$G_\infty = \sum_{i=0}^{\infty} \gamma^k = \frac{1}{1-\gamma}$$

Or $\gamma < 1$, the infinite sum has a finite value as long as the reward sequence $R_t$ is bounded
$\gamma = 0$, immediate rewards
$\gamma = 1$, normal reward.
continuing task

$$G_t = \int_0^t \gamma^\tau R_\tau d\tau$$

# Why discount

- Convenience of mathematical expression, this is also the most important
- Avoid infinite returns in MP
- Uncertainty in long-term rewards
- Immediate returns can get more benefits than delayed rewards
- Human/animal behaviour show immediate rewards are more imprtant

# Type of agents

- Model based:
  1. Explicit: model
  2. may not have policy or value function
- Model-free
  1. Explicit: policy or value functions
  2. No model

# Exploration and Exploitation

Agent only experiences what happens for the action it has tried

- Exploration: try new things that enable the agent to make better decision in the future
- Exploitation: choose actions that have good reward from past experience

Exploration-Exploitation trade-off: Sacrifice reward to explore and learn potentially better policy

# Exploration and Exploitation

Watch movies

- Exploration: watch a new movie
- Exploitation: watch a favorite movie which you have seen before

Drive a car

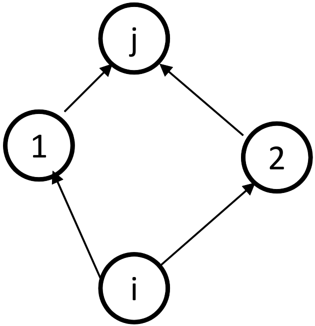
- Exploration: try a different route
- Exploitation: try fastest route given prior experience

- Evaluation: Estimate/predict the rewards from a given policy
- Control (Optimization): find the beat policy

# Markov Chains

When $n = 2$, 2-step transition probabilities (2 episodes)

$$p_{ij} = p_{i1}p_{1j}p_{i2}p_{2j}$$



# Markov Chains

Markov Chains: model of the state dynamics in the Markov property.
If it is time invariant, it is stationary, the transition probability from the state $i$ to the state $j$ is

$$\begin{aligned} & P\left(x_{k+1}=j \mid x_k=i\right) \\ = & P\left(x_{k+1}=j \mid x_k=i, x_{k-1}=i-1, \cdots, x_0=0\right) \end{aligned}$$

$n$-step transition probability

$$p_{ij}^{(n)}=p\left(x_{k+n}=j \mid x_k=i\right) * p\left(x_n=j \mid x_0=i\right)$$

# Markov Chains

Define transition matrix

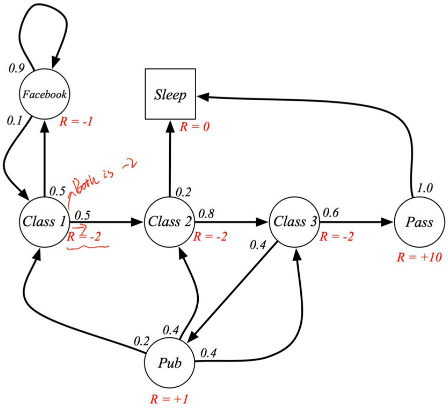
$$P = \begin{bmatrix} p\left(s_1 \mid s_1\right) & \cdots & p\left(s_N \mid s_1\right) \\ p\left(s_1 \mid s_2\right) & p\left(s_2 \mid s_2\right) & \vdots \\ p\left(s_1 \mid s_N\right) & & p\left(s_N \mid s_N\right) \end{bmatrix}$$

$p\left(s_1 \mid s_1\right)$ is the probability of $s_1 \longrightarrow s_1$
The next state is $s'$

$$s' = Ps$$

# Example

## Example: Student MRP



# Example

Return $G_t$, $\gamma = \frac{1}{2}$. $C_1$ has 4 returns.
We can see that Class 1 ($C1$) actually has 4 paths, Each path has a corresponding probability, $G_t\left(C_1\right)$ are

| | |
|---|---|
| C1 C2 C3 Pass Sleep | G1 = -2 + (-2)*1/2 + (-2)*1/4 + 10*1/8 + 0*1/16 = -2.25 |
| C1 FB FB C1 C2 Sleep | G1 = -2 + (-1)*1/2 + (-1)*1/4 + (-2)*1/8 + (-2)*1/16 + 0*1/32 = -3.125 |
| C1 C2 C3 Pub C2 C3 Pass Sleep | G1 = -2 + (-2)*1/2 + (-2)*1/4 + (1)*1/8 + (-2)*1/16 + ... = -3.41 |
| C1 FB FB C1 C2 C3 Pub C1 FB FB FB C1 C2 C3 Pub C2 Sleep | G1 = -2 + (-1)*1/2 + (-1)*1/4 + (-2)*1/8 + (-2)*1/16 + (-2)*1/32 + ... = -3.20 |

The value function needs "expected return" when evaluating the value of the state $C1$.

$$V(C1) = \frac{1}{4}[(-2.25 + (-3.125) + (-3.41) + (-3.20)]) = 2.996$$

# Computing the value function

Average return

$$\begin{aligned} V^{\pi}(s_t = s) &= \mathbb{E}\{G_t \mid s_t = s\} \\ &= \mathbb{E}_{\pi}\left[r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \cdots \mid s_t = s\right] \end{aligned}$$

Large number of episodes
Markov property yields additional structure:
Define the total discounted reward from time $t$ as the return $G_t$,

$$G_t = \sum_{i=0}^{N} \gamma^i r_{t+i}$$

where $0 \leq \gamma \leq 1$.
**Recursive form**

$$G_t = r_t + \gamma G_{t+1}$$

# Bellman equation for MRP(Markov Reward Process)

The state-value function $V(s)$ is defined as

$$V(s_t) = \mathbb{E}\left\{G_t \mid S_t = s\right\}$$

Because the recursive form $G_t = r_t + \gamma G_{t+1}$,

$$\begin{aligned} &V(s_t) = \mathbb{E}\left\{G_t \mid S_t = s\right\} \\ &= \mathbb{E}\left\{r_t + \gamma G_{t+1} \mid s\right\} \\ &= \mathbb{E}\left\{r_t + \gamma V(S_{t+1}) \mid s\right\} \\ &= r_t + \gamma \mathbb{E}\left\{V(s_{t+1})\right\} \end{aligned}$$

where $r_t$ is immediate reward, $\gamma \mathbb{E}\left\{V(s_{t+1})\right\}$ is discounted sum of future rewards

# Bellman equation for MRP

Bellman equation is

$$V(s) = r(s) + \gamma \mathbb{E}\left\{V(s')\right\}$$

where $s'$ is the state in next time.
Becasue the expectations: Averages over a time series $X$ is

$$E[A] = \sum_{a} p(A = a) A$$

The value expectation of next state $\mathbb{E}\left\{V(s')\right\}$ can be obtained according to the probability distribution of $s_t$

$$\mathbb{E}\left\{V(s')\right\} = \gamma \sum p(s' \mid s) V(s')$$

Bellman equation is

$$V(s) = r(s) + \gamma \sum p(s' \mid s) V(s')$$

# Bellman equation for MRP

Bellman equation in matrix form

$$\begin{bmatrix} V(s_1) \\ \\ V(s_N) \end{bmatrix} = \begin{bmatrix} R(s_1) \\ \\ R(s_N) \end{bmatrix} + \gamma \begin{bmatrix} P(s_1' \mid s_1) & \cdots & P(s_N' \mid s_1) \\ P(s_1' \mid s_2) & P(s_2' \mid s_2) & \vdots \\ P(s_1' \mid s_N) & & P(s_N' \mid s_N) \end{bmatrix} \Bigg[$$

For infinite horizon, the value function is stationary,

$$\begin{bmatrix} V(s_1) \\ \vdots \\ V(s_N) \end{bmatrix} = \begin{bmatrix} R(s_1) \\ \vdots \\ R(s_N) \end{bmatrix} + \gamma \begin{bmatrix} P(s_1 \mid s_1) & \cdots & P(s_N \mid s_1) \\ P(s_1 \mid s_2) & P(s_2 \mid s_2) & \vdots \\ P(s_1 \mid s_N) & & P(s_N \mid s_N) \end{bmatrix} \Bigg[$$

So

$$V = R + \gamma P V$$

$$V = (I - \gamma P)^{-1} R$$

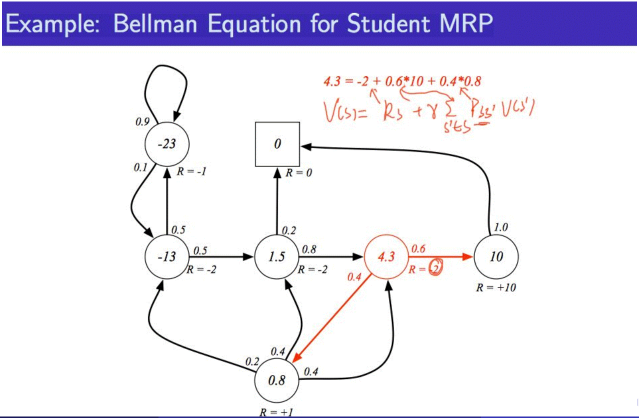
The matrix inverse require, computational complexity $O\left(N^3\right)$, $N$ states

# Value function calculation with Bellman equation

$$V(s) = r(s) + \gamma \sum p(s' \mid s)\, V(s')$$

when $\gamma = 1$

$$V(C3) = -2 + 0.6 * 10 + 0.4 * 0.8 = 4.3$$



# Markov Decision Processes (MDP)

MDP is the sets of states, actions, and rewards

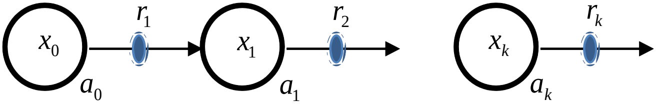
$$MDP = \{\mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{P}\}$$

$r$ is the value of the random variable $\mathcal{R}$, $r \in \mathcal{R}$
Because Markov property

$$p\left(s', r \mid s, a\right) = p\left\{s', r \mid S_{t-1} = s, A_{t-1} = a\right\}$$

The probability of each possible value for $s_t$, $r_t$ depends only on the immediately preceding state $s_{t-1}$, $a_{t-1}$

# Markov decision process




Policy at step $t$ : a mapping from states to action probabilities

$$\pi_k : x_k \longrightarrow a_k$$

or the probability that $a_k = a$ when $x_k = s$

# Markov decision process

| | Without Action | Consider Action (A) |
|---|---|---|
| Completely observable state | Markov Chain (MC) | Markov decision process (MDP) |
| Partially observable states | Hidden Markov Model (HMM) | Partially observable MDP (POMDP) |

- Factored MDPs: add planning graph style heuristics use goal regression to generalize better
- Hierarchical MDP: hierarchy of sub-tasks and/or actions to scale better
- Partially Observable Markov Decision Processes, noisy sensors, partially observable environment, popular in robotics

# Policy

Deterministic policy

$$\pi(s) = a$$

Stochastic policy

$$\pi(a \mid s) = (A_t = a \mid S_t = s)$$

Reinforcement learning: how the agent changes its policy $\pi_k$ (changing the transition probabilities), such that a better MDP (with lower cost/higher expected reward) is reached.

# MRP(Markov Reward Process)

$$MRP\left(S, R^{\pi}, P^{\pi}, \gamma\right) : MDP + \pi\left(a \mid s\right)$$

Becasue

$$V^{\pi}\left(s\right) = \mathbb{E}_{\pi}\left\{G \mid s\right\} = \mathbb{E}_{\pi}\left\{r_t + \gamma V^{\pi}\left(s'\right) \mid s\right\}$$
$$= \left(\sum_a \pi\left(a \mid s\right) \sum_{s'} \sum_r p\left(s', r \mid s, a\right)\right)\left[r_t + \gamma V^{\pi}\left(s'\right)\right]$$

where $\sum_{s'} \sum_r$are sums over all the possible values, they can be merged to $\sum_{s',r}\cdot$,

$$V^{\pi}\left(s\right) = \sum_{a,s',r} \pi\left(a \mid s\right) p\left(\bar{s}, r \mid s, a\right)\left[r_t + \gamma V\left(s'\right)\right]$$
$$= \mathbb{E}_{\pi}\left\{\pi\left(a \mid s\right)\left[r + \gamma V^{\pi}\left(\bar{s}\right)\right]\right\}$$

where

$$\mathbb{E}_{\pi}\left(r^{\pi}\left(s\right)\right) = \sum_{a \in \mathcal{A}} p\left(a \mid s\right) r\left(s, a\right)$$
$$\mathbb{E}_{\pi}\left(p^{\pi}\left(s' \mid s\right)\right) = \sum_{a \in \mathcal{A}} \pi\left(a \mid s\right) p\left(s' \mid s, a\right)$$

# Policy evaluation

Value of a policy

Initialize $V_0(s) = 0$

For $k = 1$ until convergence $(|V_k - V_{k-1}| < \varepsilon)$

For all $s$

$$V_k^{\pi}(s) = r_t(s, \pi(s)) + \gamma \sum p(s' \mid s, \pi(s)) \, V_{k-1}^{\pi}(s')$$

end

This is Bellman backup for a particular policy

# MDP control

Optimal policy

$$\pi^* (s) = \max_{\pi} [V^{\pi} (s)] \text{ for all } s \in \mathcal{S}$$

$v^* (s)$ is There exists solution
Optimal policy for a MDP in an infinite horizon problem id deterministic

# Policy Iteration (PI)

Initialize $\pi_0(s)$ randomly for all states $s$

While $i == 0$ or the policy changes for any state $(|\pi_k - \pi_{k-1}| > 0)$

Policy Evaluation

$$MRP \to V_i^{\pi}$$

Policy Improvement

$$Policty \to \pi_{i+1}$$

$i = i + 1$

# State-Action function: Q-function

This form can be also obtained from the discounted return

$$Q^{\pi}\left(s,a\right)=r\left(s,a\right)+\gamma\sum_{s'\in S}p\left(s'\mid s,a\right)V^{\pi}\left(s'\right)$$

# Policy Improvement

Compute $Q^{\pi}(s,a)$ with a policy $\pi_k$

$$Q^{\pi_k}(s,a) = r(s,a) + \gamma \sum_{s' \in S} p(s' \mid s,a) V^{\pi_k}(s') = V^{\pi}(s)$$

Compute new policy $\pi_{k+1}$ for all $s \in S$

$$\pi_{k+1}(s) = \max_a Q^{\pi_k}(s,a), \quad \forall s \in S$$

so

$$\pi_{k+1}(s) \geq Q^{\pi_k}(s, \pi_k(s))$$

# Optimal policies

For finite MDPs, an optimal policy is: a policy $\pi$ is better than or equal to a policy $\bar{\pi}$, if its expected return is greater than or equal to that of $\bar{\pi}$ for all states

$$\pi \geq \pi', \text{ if and only if } V_{\pi}\left(s\right) \geq V_{\pi'}\left(s\right) \text{ for all } s$$

Monotonic improvement in policy

**Theorem**

*For any MDP, the following points are true:*

- There is an optimal strategy that is better or at least equal to any other strategy
- All optimal strategies have the same optimal value function
- All optimal strategies have the same value function.

If a policy does not change, it will never change again

There a maximum number of interaction of policy iteration

# Bellman optimality equation

$V^{*}(s)$ also satisfies Bellman equation,

$$\begin{aligned} V^{*}(s) &= \max_{\pi}\left[V^{\pi}(s)\right] \text{ for all } s \in \mathcal{S} \\ &= \max_{a} \mathbb{E}_{\pi}\left\{r+\gamma V^{*}\left(s'\right)\right\} \\ &= \max_{a} \sum_{\bar{s},r} p\left(s', r \mid s, a\right)\left[r+\gamma V^{*}\left(s'\right)\right] \end{aligned}$$

Bellman optimality equations

$$\begin{aligned} Q^{*}(s,a) &= \max \mathbb{E}_{\pi}\left\{r(s,a)+\gamma Q\left(s', a'\right)\right\} \\ &= \mathbb{E}_{\pi}\left\{\max r(s,a)+\gamma \max_{a'} Q\left(s'.a'\right)\right\} \\ &= \sum_{\bar{s},r} p\left(\bar{s}, r \mid s, a\right)\left[\max r(s,a)+\gamma \max_{a'} Q\left(s'.a'\right)\right] \\ &= \max r(s,a)+\gamma \sum p\left(s', r \mid s, a\right) V_{\pi}^{*}\left(s'\right) \end{aligned}$$

# Value Iteration (VI)

Initialize $V_0(s) = 0$ randomly for all states $s$
Loop until finite horizon or convergence
    for each state

$$V_{k+1}(s) = \max_a r(s, a) + \gamma \sum_{s' \in S} p(s' \mid s, a) V_k(s')$$

Bellman backup on value function

$$\pi_{k+1}(s) = \arg\max_a \left\{ r(s, a) + \gamma \sum_{s' \in S} p(s' \mid s, a) V_k(s') \right\}$$