Regret Bounds for Optimistic Algorithms in Multi-armed Bandits and MDPs

Ronald Ortner

Montanuniversität Leoben

Chalmers MLSS 2015

Goal: For given problem, find algorithm that

- works well in practice (performance, efficiency),
- has favorable theoritical guarantees.

Goal: For given problem, find algorithm that

- works well in practice (performance, efficiency),
- has favorable theoritical guarantees.

If problem is too hard, you can start from either end:

Goal: For given problem, find algorithm that

- works well in practice (performance, efficiency),
- has favorable theoritical guarantees.

If problem is too hard, you can start from either end:

1. engineer algorithm that works well in practice
   - works maybe only for a particular case
   - often not quite clear why it works

Goal: For given problem, find algorithm that

- works well in practice (performance, efficiency),
- has favorable theoritical guarantees.

If problem is too hard, you can start from either end:

1. engineer algorithm that works well in practice
   - works maybe only for a particular case
   - often not quite clear why it works
2. create algorithm for which you can show theoretical bounds
   - works maybe only under additional assumptions
   - often computationally inefficient

Machine Learning: Two Extremes

If problem is too hard, you can start from either end:

1. engineer algorithm that works well in practice
2. create algorithm for which you can show theoretical bounds

What is common to both approaches:
you want to do better than all others
(better performance, better bounds)

General Setting:

- At each step $t = 1, 2, \ldots, T$
  - observe state s_t,
  - choose an action a_t from a given action set A,
  - receive reward r_t (might be random, typically depends on s_t and a_t).

General Setting:

- At each step $t = 1, 2, \ldots, T$
  - observe state s_t,
  - choose an action a_t from a given action set A,
  - receive reward r_t (might be random, typically depends on s_t and a_t).
- **Goal:** maximize collected reward $\sum_{t=1}^{T} r_t$.

General Setting:

- At each step $t = 1, 2, \ldots, T$
  - observe state s_t,
  - choose an action a_t from a given action set A,
  - receive reward r_t (might be random, typically depends on s_t and a_t).
- **Goal:** maximize collected reward $\sum_{t=1}^{T} r_t$.
- Observations and rewards are generated by an unknown environment.

General Setting:

- At each step $t = 1, 2, \ldots, T$
  - observe state s_t,
  - choose an action a_t from a given action set A,
  - receive reward r_t (might be random, typically depends on s_t and a_t).
- **Goal:** maximize collected reward $\sum_{t=1}^{T} r_t$.
- Observations and rewards are generated by an unknown environment.
- States, actions, and rewards is the only information available to the learner.

General Setting:

- At each step $t = 1, 2, \ldots, T$
  - observe state s_t,
  - choose an action a_t from a given action set A,
  - receive reward r_t (might be random, typically depends on s_t and a_t).
- **Goal:** maximize collected reward $\sum_{t=1}^{T} r_t$.
- Observations and rewards are generated by an unknown environment.
- States, actions, and rewards is the only information available to the learner.
- $\rightsquigarrow$ Policies map histories to action, i.e.

$$a_t = \pi(s_1, a_1, r_1, \ldots, s_{t-1}, a_{t-1}, r_{t-1}).$$

General Setting:

- At each step $t = 1, 2, \ldots, T$
  - observe state s_t,
  - choose an action a_t from a given action set A,
  - receive reward r_t (might be random, typically depends on s_t and a_t).
- **Goal:** maximize collected reward $\sum_{t=1}^{T} r_t$.
- $\rightsquigarrow$ Policies map histories to action, i.e.

$$a_t = \pi(s_1, a_1, r_1, \ldots, s_{t-1}, a_{t-1}, r_{t-1}).$$

General Setting:

- At each step $t = 1, 2, \ldots, T$
  - observe state s_t,
  - choose an action a_t from a given action set A,
  - receive reward r_t (might be random, typically depends on s_t and a_t).
- **Goal:** maximize collected reward $\sum_{t=1}^{T} r_t$.
- $\rightsquigarrow$ Policies map histories to action, i.e.

$$a_t = \pi(s_1, a_1, r_1, \ldots, s_{t-1}, a_{t-1}, r_{t-1}).$$

- If environment were known, optimal policy would be π^*.

General Setting:

- At each step $t = 1, 2, \ldots, T$
  - observe state s_t,
  - choose an action a_t from a given action set A,
  - receive reward r_t (might be random, typically depends on s_t and a_t).
- **Goal:** maximize collected reward $\sum_{t=1}^{T} r_t$.
- $\rightsquigarrow$ Policies map histories to action, i.e.

$$a_t = \pi(s_1, a_1, r_1, \ldots, s_{t-1}, a_{t-1}, r_{t-1}).$$

- If environment were known, optimal policy would be π^*.
- We'd like to have algorithms with theoretical performance guarantees when compared to the optimal policy π^*.

- At each step $t = 1, 2, \ldots, T$ we observe option (secretary) s_t.
- Any option s_t gives (deterministic) reward $r(s_t)$.

- At each step $t = 1, 2, \ldots, T$ we observe option (secretary) s_t.
- Any option s_t gives (deterministic) reward $r(s_t)$.
- At any step we can either choose the current option s_t ...
    - $\rightsquigarrow$ receive reward $r(s_t)$ and quit

- At each step $t = 1, 2, \ldots, T$ we observe option (secretary) s_t.
- Any option s_t gives (deterministic) reward $r(s_t)$.
- At any step we can either choose the current option s_t …
  - $\rightsquigarrow$ receive reward $r(s_t)$ and quit
- … or continue to see the next option.

- At each step $t = 1, 2, \ldots, T$ we observe option (secretary) s_t.
- Any option s_t gives (deterministic) reward $r(s_t)$.
- At any step we can either choose the current option s_t ...
  - $\rightsquigarrow$ receive reward $r(s_t)$ and quit
- ... or continue to see the next option.
- **Goal:** Choose the best option.

Outline

1. Multi-armed bandit problems
   - Introduction
   - Algorithms
   - Analysis
2. Markov decision processes
   - Introduction
   - An Optimistic Algorithm for RL in MDPs
   - Regret Bounds
3. Outlook
   - Colored MDPs
   - From Colored to Continuous State MDPs
   - UCRL2 revisited: Bias and Diameter
   - Continuous State MDPs

Multi-armed bandits

Setting:

- At times $t = 1, 2, \ldots$ choose an arm a_t from a finite set of arms A.
- receive for chosen arm a random reward $\in [0, 1]$ with mean $r(a)$.

Setting:

- Learner chooses at times $t = 1, 2, \ldots$ an arm a_t from a finite set of arms A.
- receives for chosen arm a random reward $\in [0, 1]$ with mean $r(a)$.

Goal(s):

Setting:

- Learner chooses at times $t = 1, 2, \ldots$ an arm a_t from a finite set of arms A.
- receives for chosen arm a random reward $\in [0, 1]$ with mean $r(a)$.

Goal(s):

- Identify optimal arm a^* with maximal reward r^*.

Setting:

- Learner chooses at times $t = 1, 2, \ldots$ an arm a_t from a finite set of arms A.
- receives for chosen arm a random reward $\in [0, 1]$ with mean $r(a)$.

Goal(s):

- Identify optimal arm a^* with maximal reward r^*.
- Do this in an online fashion, so that collected rewards $\sum_{t=1}^{T} r_t$ are maximized, where r_t is the random reward at step t.

Setting:

- Learner chooses at times $t = 1, 2, \ldots$ an arm a_t from a finite set of arms A.
- receives for chosen arm a random reward $\in [0, 1]$ with mean $r(a)$.

Goal(s):

- Identify optimal arm a^* with maximal reward r^*.
- Do this in an online fashion, so that collected rewards $\sum_{t=1}^{T} r_t$ are maximized, where r_t is the random reward at step t.
  (This shall happen for all T, not just for $T \to \infty$.)

Setting:

- Learner chooses at times $t = 1, 2, \ldots$ an arm a_t from a finite set of arms A.
- receives for chosen arm a random reward $\in [0, 1]$ with mean $r(a)$.

Goal(s):

- Identify optimal arm a^* with maximal reward r^*.
- Do this in an online fashion, so that collected rewards $\sum_{t=1}^{T} r_t$ are maximized, where r_t is the random reward at step t.
  (This shall happen for all T, not just for $T \to \infty$.)
- $\rightsquigarrow$ Minimize the T-step regret

$$Tr^* - \sum_{t=1}^{T} r_t.$$

Minimize the *T-step regret* $Tr^* - \sum_{t=1}^{T} r_t$.

What bounds on the regret can we expect?

Minimize the T-step regret $Tr^* - \sum_{t=1}^{T} r_t$.

What bounds on the regret can we expect?

- Playing a suboptimal arm a all the time has linear regret

$$T(r^* - r(a)).$$

Minimize the T-step regret $Tr^* - \sum_{t=1}^{T} r_t$.

What bounds on the regret can we expect?

- Playing a suboptimal arm a all the time has linear regret

$$T(r^* - r(a)).$$

- For $T \to \infty$, we'd expect a good algorithm to identify the optimal arm, so that the per-step regret

$$\lim_{T\to\infty} \frac{Tr^* - \sum_{t=1}^{T} r_t}{T} = 0.$$

Minimize the T-step regret $Tr^* - \sum_{t=1}^{T} r_t$.

What bounds on the regret can we expect?

- Playing a suboptimal arm a all the time has linear regret

$$T(r^* - r(a)).$$

- For $T \to \infty$, we'd expect a good algorithm to identify the optimal arm, so that the per-step regret

$$\lim_{T\to\infty} \frac{Tr^* - \sum_{t=1}^{T} r_t}{T} = 0.$$

- $\rightsquigarrow$ We'd expect a good algorithm to have sublinear regret.

Minimize the T-step regret $Tr^* - \sum_{t=1}^{T} r_t$.

What bounds on the regret can we expect?

- Playing a suboptimal arm a all the time has linear regret

$$T(r^* - r(a)).$$

- For $T \to \infty$, we'd expect a good algorithm to identify the optimal arm, so that the per-step regret

$$\lim_{T \to \infty} \frac{Tr^* - \sum_{t=1}^{T} r_t}{T} = 0.$$

- $\rightsquigarrow$ We'd expect a good algorithm to have sublinear regret.
- The smaller the regret rate, the faster the algorithm converges to the optimal solution.

- ***Experiment design:***
  - different treatments for disease
  - would like to use best treatment

- ***Experiment design:***
  - different treatments for disease
  - would like to use best treatment
- ***Routing in networks:***

  look for shortest path in network

- ***Experiment design:***
  - different treatments for disease
  - would like to use best treatment
- ***Routing in networks:***

  look for shortest path in network
- ***Pricing:***

  would like to sell for the highest price for which a customer is willing to buy

- ***Experiment design:***
  - different treatments for disease
  - would like to use best treatment
- ***Routing in networks:***

  look for shortest path in network
- ***Pricing:***

  would like to sell for the highest price for which a customer is willing to buy
- ***Placing ads on webpages:***
  - different ads or different pics for the same product
  - would like to use the one which is most likely to be clicked at

A simple algorithm:

- Choose each arm once.
- Always choose the arm with the best mean reward so far.

A simple algorithm:

- Choose each arm once.
- Always choose the arm with the best mean reward so far.

Problem:
In case the optimal arm a^* gets low reward at the beginning, it wouldn't be chosen anymore.

$\rightarrow$ ***"Exploration vs. Exploitation" dilemma:***

A simple algorithm:

- Choose each arm once.
- Always choose the arm with the best mean reward so far.

Problem:
In case the optimal arm a^* gets low reward at the beginning,
it wouldn't be chosen anymore.

$\rightarrow$ ***"Exploration vs. Exploitation" dilemma:***

- Play best arm so far, ...

A simple algorithm:

- Choose each arm once.
- Always choose the arm with the best mean reward so far.

Problem:
In case the optimal arm a^* gets low reward at the beginning, it wouldn't be chosen anymore.

$\rightarrow$ ***"Exploration vs. Exploitation" dilemma:***

- Play best arm so far, ...
- ... or rather explore a different arm?

A simple algorithm:

- Choose each arm once.
- Always choose the arm with the best mean reward so far.

Problem:
In case the optimal arm a^* gets low reward at the beginning, it wouldn't be chosen anymore.

$\rightarrow$ ***"Exploration vs. Exploitation" dilemma:***

Possible solution ("ε-greedy"):
With small probability ε choose a different arm.

- The empirical mean

$$\hat{r}(a) := \frac{\text{total reward for arm } a}{\text{number of plays of arm } a}$$

gives an estimate for real mean reward $r(a)$.

- The empirical mean

$$\hat{r}(a) := \frac{\text{total reward for arm } a}{\text{number of plays of arm } a}$$

gives an estimate for real mean reward $r(a)$.

▷ How good is this estimate?

- The empirical mean

$$\hat{r}(a) := \frac{\text{total reward for arm } a}{\text{number of plays of arm } a}$$

gives an estimate for real mean reward $r(a)$.

▷ How good is this estimate?

- Use confidence intervals to get answer with high probability:

- The empirical mean

$$\hat{r}(a) := \frac{\text{total reward for arm } a}{\text{number of plays of arm } a}$$

gives an estimate for real mean reward $r(a)$.

▷ How good is this estimate?

- Use confidence intervals to get answer with high probability:

E.g., Chernov-Hoeffding bound

With probability $\geq 1 - \delta$ the true mean $r(a)$ is contained in confidence interval

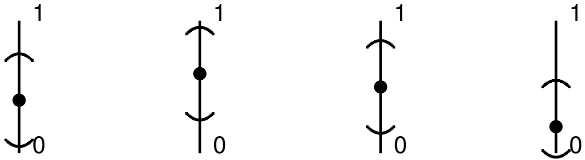
$$\left[\hat{r}(a) - \sqrt{\frac{\log\frac{2}{\delta}}{2n}}, \hat{r}(a) + \sqrt{\frac{\log\frac{2}{\delta}}{2n}}\right],$$

where n is the number of samples.

(n counts how often an arm a has been played.)

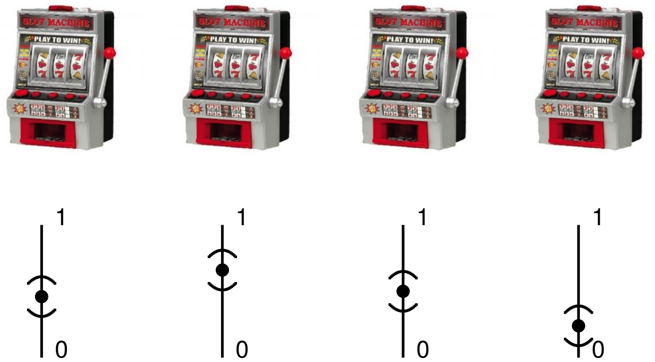
Improved algorithm:

- Choose arms alternatingly.
- Eliminate arm, if its confidence interval is below the confidence interval of another arm.



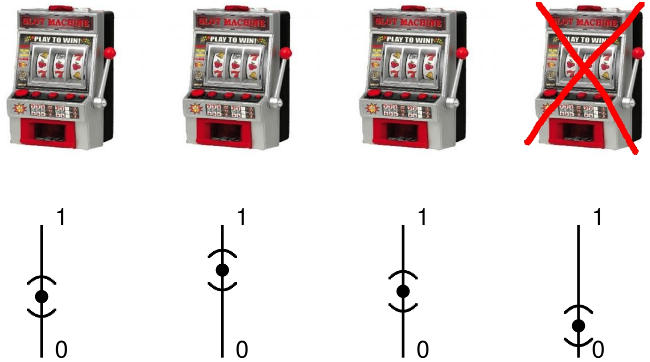
Improved algorithm:

- Choose arms alternatingly.
- Eliminate arm, if its confidence interval is below the confidence interval of another arm.



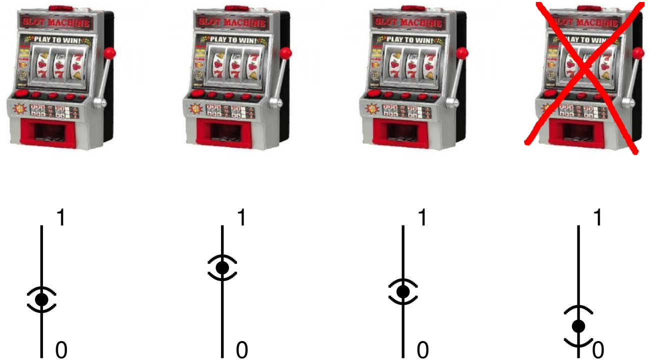
Improved algorithm:

- Choose arms alternatingly.
- Eliminate arm, if its confidence interval is below the confidence interval of another arm.



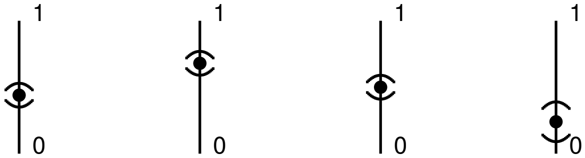
Improved algorithm:

- Choose arms alternatingly.
- Eliminate arm, if its confidence interval is below the confidence interval of another arm.



Improved algorithm:

- Choose arms alternatingly.
- Eliminate arm, if its confidence interval is below the confidence interval of another arm.



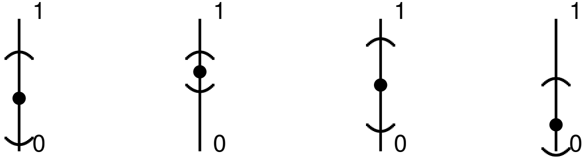
Improved algorithm:

- Choose arms alternatingly.
- Eliminate arm, if its confidence interval is below the confidence interval of another arm.

Problem:
Suboptimal arm is played relatively often, and even if confidence intervals are hardly intersecting.
($\rightarrow$ Play a lot of arms that are suboptimal w.h.p.)

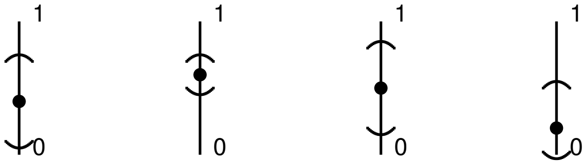
Optimistic algorithm UCB (Auer, Fischer, Cesa-Bianchi 2002)

- Choose each arm once.
- Choose arm with maximal upper confidence bound.



Optimistic algorithm UCB (Auer, Fischer, Cesa-Bianchi 2002)

- Choose each arm once.
- Choose arm with maximal upper confidence bound.



Optimistic algorithm UCB (Auer, Fischer, Cesa-Bianchi 2002)

- Choose each arm once.
- Choose arm with maximal upper confidence bound.

Optimistic algorithm UCB (Auer, Fischer, Cesa-Bianchi 2002)

- Choose each arm once.
- Choose arm with maximal upper confidence bound.

Idea:

- Either get high reward ($\to$ good), or

Optimistic algorithm UCB (Auer, Fischer, Cesa-Bianchi 2002)

- Choose each arm once.
- Choose arm with maximal upper confidence bound.

Idea:

- Either get high reward ($\rightarrow$ good), or
- get low reward ($\rightarrow$ but learn something).

Optimistic algorithm UCB (Auer, Fischer, Cesa-Bianchi 2002)

- Choose each arm once.
- Choose arm with maximal upper confidence bound.

Choice of confidence intervals:

Optimistic algorithm UCB (Auer, Fischer, Cesa-Bianchi 2002)

- Choose each arm once.
- Choose arm with maximal upper confidence bound.

Choice of confidence intervals:

- If error probability of confidence intervals is fixed, the error probability becomes arbitrarily large.

Optimistic algorithm UCB (Auer, Fischer, Cesa-Bianchi 2002)

- Choose each arm once.
- Choose arm with maximal upper confidence bound.

Choice of confidence intervals:

- If error probability of confidence intervals is fixed, the error probability becomes arbitrarily large.
- $\rightsquigarrow$ Choose confidence intervals so that sum of error probabilities over all time steps remains bounded.

UCB Algorithmus (Auer, Fischer, Cesa-Bianchi 2002)

- Choose each arm once.
- Choose arm with maximal upper confidence bound, that is, at step t choose

$$\arg\max_{a \in A} \left\{ \hat{r}_t(a) + \mathrm{conf}_t(a) \right\}.$$

Choice of confidence intervals:

- For $\mathrm{conf}_t(a) = \sqrt{\frac{2\log(t/\delta)}{n_t(a)}}$, the error probability for the confidence interval of one arm is $\frac{\delta}{t^4}$ (Chernov-Hoeffding).
- In this case, the sum over all error probabilities is $\leq \delta$.

Consider an arbitrary suboptimal arm a.
We ignore the randomness of the reward for a and consider the pseudoregret

$$\begin{aligned}
&\sum_{t:a_t=a} \left(r^* - r(a)\right) \\
&\quad = \sum_{t:a_t=a} (r^* - (\hat{r}_t(a) + \mathrm{conf}_t(a))) \\
&\qquad + \sum_{t:a_t=a} ((\hat{r}_t(a) + \mathrm{conf}_t(a)) - r(a))
\end{aligned}$$

Consider an arbitrary suboptimal arm a.
We ignore the randomness of the reward for a and consider the pseudoregret

$$
\begin{aligned}
&\sum_{t:a_t=a} \left(r^* - r(a)\right) \\
&= \sum_{t:a_t=a} (r^* - (\hat{r}_t(a) + \mathrm{conf}_t(a))) \\
&\quad + \sum_{t:a_t=a} ((\hat{r}_t(a) + \mathrm{conf}_t(a)) - r(a))
\end{aligned}
$$

First term: ≤ 0, since w.h.p. $\hat{r}_t(a) + \mathrm{conf}_t(a) \geq \hat{r}_t(a^*) + \mathrm{conf}_t(a^*) \geq r^*$

$$\sum_{t:a_t=a} \Big(r^* - r(a)\Big)$$

$$\leq \sum_{t:a_t=a} ((\hat{r}_t(a) + \mathrm{conf}_t(a)) - r(a))$$

$$
\begin{aligned}
&\sum_{t:a_t=a} \left(r^* - r(a)\right) \\
&\quad \leq \sum_{t:a_t=a} ((\hat{r}_t(a) + \mathrm{conf}_t(a)) - r(a)) \\
&\quad = \sum_{t:a_t=a} \left(\hat{r}_t(a) - r(a)\right) + \sum_{t:a_t=a} \mathrm{conf}_t(a)
\end{aligned}
$$

$$
\begin{aligned}
&\sum_{t:a_t=a} \Big(r^* - r(a)\Big) \\
&\leq \sum_{t:a_t=a} ((\hat{r}_t(a) + \mathrm{conf}_t(a)) - r(a)) \\
&= \sum_{t:a_t=a} \Big(\hat{r}_t(a) - r(a)\Big) + \sum_{t:a_t=a} \mathrm{conf}_t(a) \\
&\leq 2 \sum_{t:a_t=a} \mathrm{conf}_t(a_t) \leq 2\sqrt{2 \log \frac{T}{\delta}} \cdot \sum_{t:a_t=a} \frac{1}{\sqrt{n_t(a)}}
\end{aligned}
$$

$$
\begin{aligned}
\sum_{t:a_t=a} \left(r^* - r(a)\right) \\
&\le \sum_{t:a_t=a} \left((\hat{r}_t(a) + \mathrm{conf}_t(a)) - r(a)\right) \\
&= \sum_{t:a_t=a} \left(\hat{r}_t(a) - r(a)\right) + \sum_{t:a_t=a} \mathrm{conf}_t(a) \\
&\le 2 \sum_{t:a_t=a} \mathrm{conf}_t(a_t) \le 2\sqrt{2 \log \tfrac{T}{\delta}} \cdot \sum_{t:a_t=a} \frac{1}{\sqrt{n_t(a)}} \\
&= 2\sqrt{2 \log \tfrac{T}{\delta}} \cdot \sum_{t=1}^{n_T(a)} \frac{1}{\sqrt{t}}
\end{aligned}
$$

$$
\begin{aligned}
&\sum_{t:a_t=a} \Big(r^* - r(a)\Big) \\
&\le \sum_{t:a_t=a} \big((\hat{r}_t(a) + \mathrm{conf}_t(a)) - r(a)\big) \\
&= \sum_{t:a_t=a} \Big(\hat{r}_t(a) - r(a)\Big) + \sum_{t:a_t=a} \mathrm{conf}_t(a) \\
&\le 2 \sum_{t:a_t=a} \mathrm{conf}_t(a_t) \le 2\sqrt{2\log\tfrac{T}{\delta}} \cdot \sum_{t:a_t=a} \frac{1}{\sqrt{n_t(a)}} \\
&= 2\sqrt{2\log\tfrac{T}{\delta}} \cdot \sum_{t=1}^{n_T(a)} \frac{1}{\sqrt{t}} \le 4\sqrt{2\log\tfrac{T}{\delta}} \cdot \sqrt{n_T(a)}
\end{aligned}
$$

The pseudoregret w.r.t. an arbitrary suboptimal arm a is

$$\sum_{t:a_t=a} \Big(r^* - r(a)\Big) \;\leq\; 4\sqrt{2\log\tfrac{T}{\delta}} \cdot \sqrt{n_T(a)}.$$

Summing over all suboptimal arms, Jensen's inequality gives

Theorem

With probability at least $1-\delta$ the pseudoregret of UCB is bounded as

$$\sum_{t=1}^{T} \Big(r^* - r(a_t)\Big) \leq 4\sqrt{2|A|T\log\tfrac{T}{\delta}}.$$

Theorem

For a real convex function φ, numbers $x_1, x_2, \ldots, x_n$, and positive weights a_i, it holds that

$$\varphi\left(\frac{\sum a_i x_i}{\sum a_i}\right) \leq \frac{\sum a_i \varphi(x_i)}{\sum a_i}.$$

On the other hand, if φ is concave, we have

$$\varphi\left(\frac{\sum a_i x_i}{\sum a_i}\right) \geq \frac{\sum a_i \varphi(x_i)}{\sum a_i}.$$

$$\Rightarrow \quad \sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i} \geq \frac{1}{n}\sum_{i=1}^{n}\sqrt{x_i} \quad \Rightarrow \quad \sqrt{n\sum_{i=1}^{n} x_i} \geq \sum_{i=1}^{n}\sqrt{x_i}$$

The regret w.r.t. an arbitrary suboptimal arm *a* is

$$\sum_{t:a_t=a} \Big(r^* - r(a)\Big) \;\le\; 4\sqrt{2\log\tfrac{T}{\delta}} \cdot \sqrt{n_T(a)}.$$

Summing over all suboptimal arms, Jensen's inequality gives

Theorem

With probability at least $1-\delta$ the pseudoregret of UCB is bounded as

$$\sum_{t=1}^{T} \Big(r^* - r(a_t)\Big) \le 4\sqrt{2|A|T\log\tfrac{T}{\delta}}.$$

Now we assume that $\mathrm{conf}_t(a) := \sqrt{\frac{2\log t}{n_t(a)}}$ and take a look at the expected regret.

By Wald's lemma we can write it as

$$\mathbb{E}\left[\sum_{t=1}^{T}\Big(r^* - r(a_t)\Big)\right] = \sum_{a:r(a)<r^*} \mathbb{E}[n_T(a)] \cdot (r^* - r(a)).$$

Hence, we'd like to bound $\mathbb{E}[n_T(a)]$ for suboptimal arms a.

Theorem

Let $X_1, X_2, \ldots$ be a sequence of i.i.d. random variables, and let N be a nonnegative random integer that is independent of the sequence $X_1, X_2, \ldots$.

If N and the X_i have finite expectations, then

$$\mathbb{E}[X_1 + \cdots + X_N] = \mathbb{E}[N] \cdot \mathbb{E}[X_1] \, .$$

Now we assume that $\mathrm{conf}_t(a) := \sqrt{\frac{2\log t}{n_t(a)}}$ and take a look at the expected regret.

By Wald's lemma we can write it as

$$\mathbb{E}\left[\sum_{t=1}^{T}\Big(r^* - r(a_t)\Big)\right] = \sum_{a:r(a)<r^*} \mathbb{E}[n_T(a)] \cdot (r^* - r(a)).$$

Hence, we'd like to bound $\mathbb{E}[n_T(a)]$ for suboptimal arms a.

Consider a suboptimal arm a and its difference $r^* - r(a)$ to the optimal arm.

Consider a suboptimal arm a and its difference $r^* - r(a)$ to the optimal arm.
When a has been played

$$n_t(a) \approx \frac{8 \log T}{(r^* - r(a))^2}$$

times, then $\sqrt{\frac{2 \log t}{n_t(a)}} = \frac{(r^* - r(a))}{2}$, so that by Chernov-Hoeffding w.h.p.

$$\hat{r}(a) + \sqrt{\frac{2 \log t}{n_t(a)}} \leq r(a) + 2\sqrt{\frac{2 \log t}{n_t(a)}} = r(a) + 2 \cdot \frac{(r^* - r(a))}{2} = r^*.$$

Consider a suboptimal arm a and its difference $r^* - r(a)$ to the optimal arm.
When a has been played

$$n_t(a) \approx \frac{8 \log T}{(r^* - r(a))^2}$$

times, then $\sqrt{\frac{2 \log t}{n_t(a)}} = \frac{(r^* - r(a))}{2}$, so that by Chernov-Hoeffding w.h.p.

$$\hat{r}(a) + \sqrt{\frac{2 \log t}{n_t(a)}} \leq r(a) + 2\sqrt{\frac{2 \log t}{n_t(a)}} = r(a) + 2 \cdot \frac{(r^* - r(a))}{2} = r^*.$$

Hence w.h.p.

$$\hat{r}(a^*) + \sqrt{\frac{2 \log t}{n_t(a^*)}} \geq r(a^*) > \hat{r}(a) + \sqrt{\frac{2 \log t}{n_t(a)}},$$

and UCB doesn't play arm a anymore.

Theorem (Auer et al., 2002a)

The expected number of times a suboptimal arm a is chosen is bounded as

$$\mathbb{E}[n_T(a)] \leq \frac{8 \log T}{(r^* - r(a))^2}.$$

Hence, the expected regret of UCB is bounded as

$$\mathbb{E}\left[\sum_{t=1}^{T} \left(r^* - r(a_t)\right)\right] \leq \sum_{a : r(a) < r^*} \frac{8 \log T}{r^* - r(a)}.$$

Which one is better?

Regret Bound I for UCB

$$\sum_{t=1}^{T}\left(r^* - r(a_t)\right) \leq 4\sqrt{2|A|T\log T}.$$

Regret Bound II for UCB

$$\mathbb{E}\left[\sum_{t=1}^{T}\left(r^* - r(a_t)\right)\right] \leq \sum_{a:r(a)<r^*} \frac{8\log T}{r^* - r(a)}.$$

Which one is better?

Regret Bound I for UCB

$$\sum_{t=1}^{T} \left(r^* - r(a_t) \right) \leq 4\sqrt{2|A| T \log T}.$$

Regret Bound II for UCB

$$\mathbb{E}\left[\sum_{t=1}^{T} \left(r^* - r(a_t) \right) \right] \leq \sum_{a: r(a) < r^*} \frac{8 \log T}{r^* - r(a)}.$$

This depends on the distance $r^* - r(a)$!

Which one is better?

Regret Bound I for UCB

$$\sum_{t=1}^{T} \left(r^* - r(a_t) \right) \leq 4\sqrt{2|A| T \log T}.$$

Regret Bound II for UCB

$$\mathbb{E}\left[\sum_{t=1}^{T} \left(r^* - r(a_t) \right) \right] \leq \sum_{a : r(a) < r^*} \frac{8 \log T}{r^* - r(a)}.$$

This depends on the distance $r^* - r(a)$!
E.g., for $r^* - r(a) < 1/\sqrt{T}$, the first bound is better.

Theorem (Auer et al., 2002b)

For any K and any T there exists a bandit problem with K arms such that the expected regret of any algorithm after T steps is at least

$$\text{const} \cdot \sqrt{KT}.$$

Theorem (Mannor & Tsitsiklis, 2004)

For any K there exists a bandit problem with K arms such that for any T the expected regret of any algorithm is at least

$$\text{const} \cdot \sum_{a:r(a)<r^*} \frac{\log(T(r^* - r(a))/K)}{r^* - r(a)}.$$

Outline

1. Multi-armed bandit problems
   - Introduction
   - Algorithms
   - Analysis
2. Markov decision processes
   - Introduction
   - An Optimistic Algorithm for RL in MDPs
   - Regret Bounds
3. Outlook
   - Colored MDPs
   - From Colored to Continuous State MDPs
   - UCRL2 revisited: Bias and Diameter
   - Continuous State MDPs

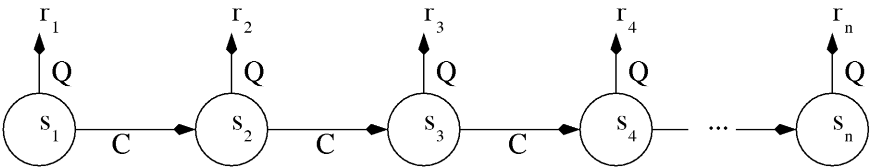
Definition

Markov decision process (MDP) $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, s_1, p, r \rangle$:

$\mathcal{S}$	$\ldots$	state space
$\mathcal{A}$	$\ldots$	a set of actions available in each state

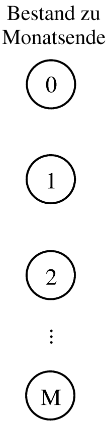
- Start in initial state s_1.
- When choosing action a in state s:
  - ▷ random reward with mean $r(s, a)$ in $[0, 1]$,
  - ▷ transition to state s' with probability $p(s'|s, a)$.

- At each step $t = 1, 2, \ldots, T$ we observe option (secretary) s_t.
- Any option s_t gives (deterministic) reward $r(s_t)$.
- At any step we can either choose the current option s_t ...
  - $\rightsquigarrow$ receive reward $r(s_t)$ and quit
- ... or continue to see the next option.
- **Goal:** Choose the best option.

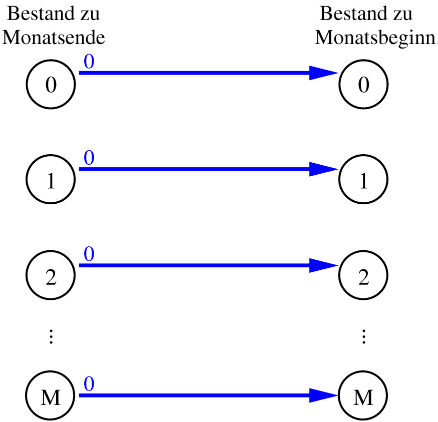


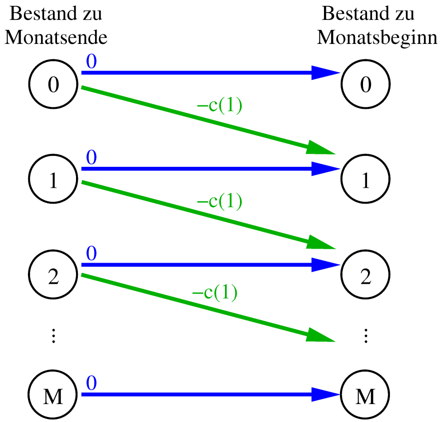
Inventory management in a warehouse:

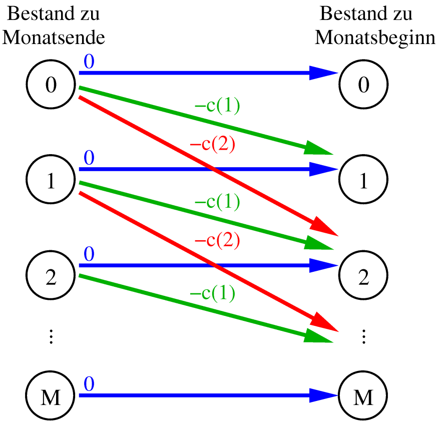
- At the end of month one looks at current inventory...
- ... and submit orders.
- Demand is random.
- There are costs for storing goods.

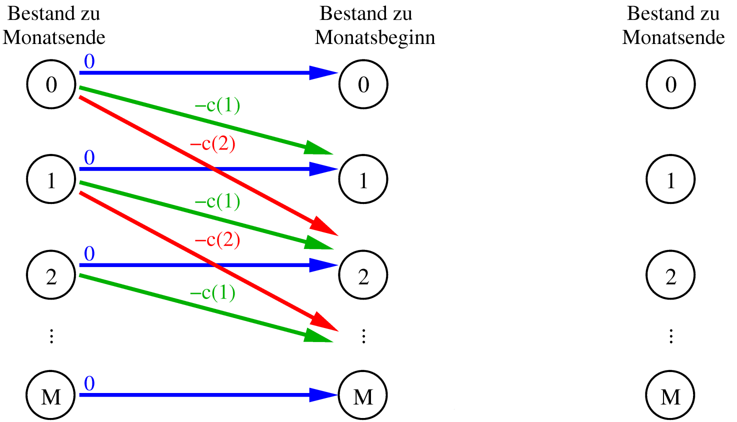

Bestand zu
Monatsende
0
1
2
M

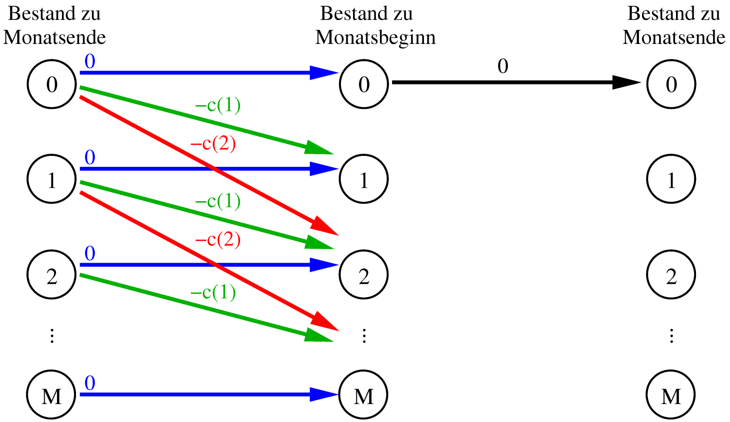
MDPs: Another Example for Illustration

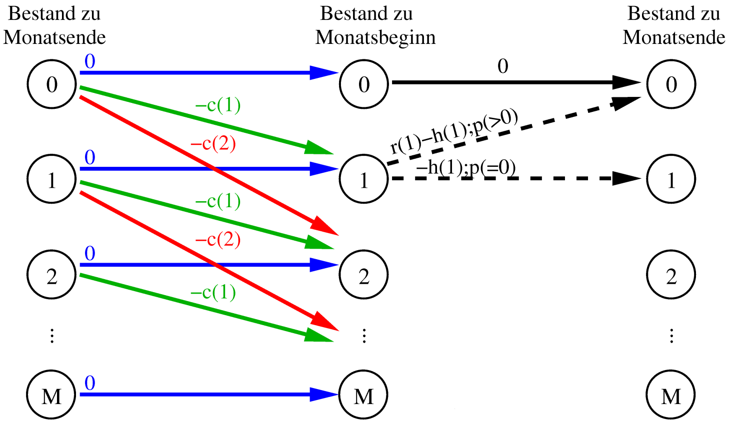


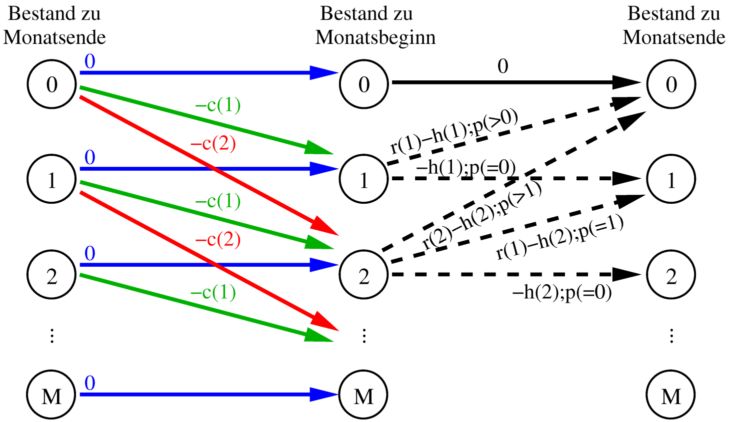

Bestand zu
Monatsende
Bestand zu
Monatsbeginn
0
0
0
−c(1)
1
0
1
−c(1)
2
0
2
−c(1)
⋮
⋮
M
0
M

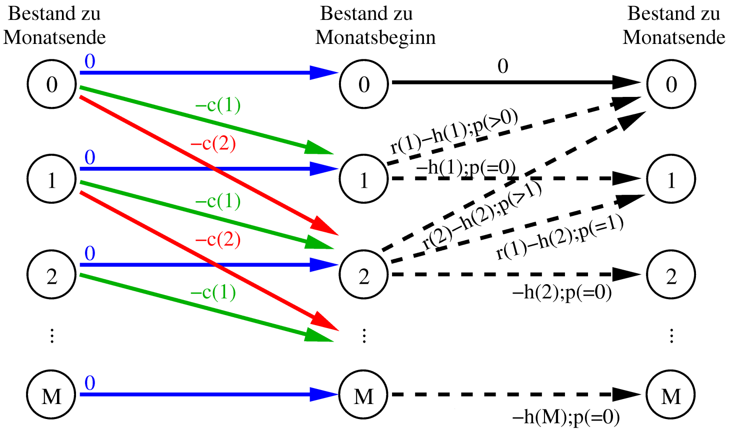

Bestand zu
Monatsende
Bestand zu
Monatsbeginn
0
1
2
⋮
M
0
1
2
⋮
M
0
0
0
0
−c(1)
−c(2)
−c(1)
−c(2)
−c(1)


Bestand zu Monatsende
Bestand zu Monatsbeginn
Bestand zu Monatsende
0
1
2
M
0
1
2
M
0
1
2
M
0
0
0
0
−c(1)
−c(2)
−c(1)
−c(2)
−c(1)


Bestand zu Monatsende
Bestand zu Monatsbeginn
Bestand zu Monatsende
0
0
0
0
1
1
1
2
2
2
M
M
M
0
0
0
0
0
−c(1)
−c(2)
−c(1)
−c(2)
−c(1)


Bestand zu Monatsende
Bestand zu Monatsbeginn
Bestand zu Monatsende
0
1
2
M
0
1
2
M
0
1
2
M
0
−c(1)
−c(2)
0
−c(1)
−c(2)
0
−c(1)
0
0
r(1)−h(1);p(>0)
−h(1);p(=0)


Bestand zu Monatsende
Bestand zu Monatsbeginn
Bestand zu Monatsende
0
1
2
M
0
1
2
M
0
1
2
M
0
0
0
0
0
−c(1)
−c(2)
−c(1)
−c(2)
−c(1)
r(1)−h(1);p(>0)
−h(1);p(=0)
r(2)−h(2);p(>1)
r(1)−h(2);p(=1)
−h(2);p(=0)


Bestand zu Monatsende
Bestand zu Monatsbeginn
Bestand zu Monatsende
0
1
2
M
0
−c(1)
−c(2)
0
−c(1)
−c(2)
0
−c(1)
0
r(1)−h(1);p(>0)
−h(1);p(=0)
r(2)−h(2);p(>1)
r(1)−h(2);p(=1)
−h(2);p(=0)
−h(M);p(=0)

- *1940s:* dynamic programming for optimization problems
  (Richard Bellman)

- *1940s:* dynamic programming for optimization problems (Richard Bellman)

Basic idea:

Any optimal solution of a problem induces an optimal solution for a subproblem.

- *1940s:* dynamic programming for optimization problems (Richard Bellman)

Basic idea:

Any optimal solution of a problem induces an optimal solution for a subproblem.

$\rightsquigarrow$ Bellman equation

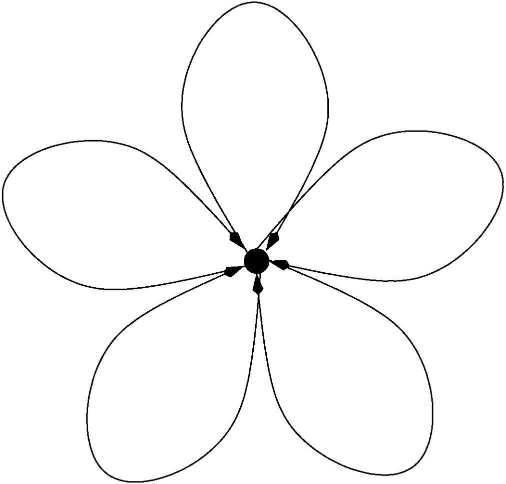
- *1940s:* dynamic programming for optimization problems
  (Richard Bellman)

- *1940s:* dynamic programming for optimization problems (Richard Bellman)
- *1950s: stochastic* dynamic programming
  $\rightarrow$ *MDPs* (Richard Bellman)

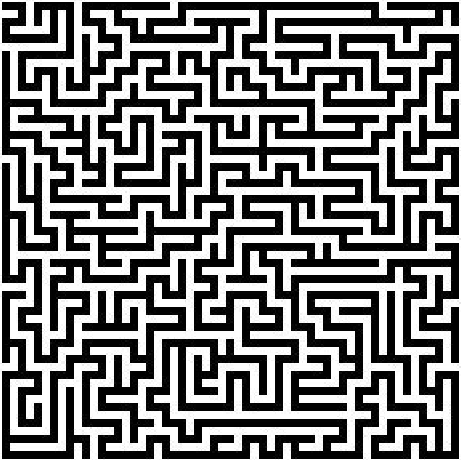
- *1940s:* dynamic programming for optimization problems (Richard Bellman)
- *1950s: stochastic* dynamic programming
  $\rightarrow$ *MDPs* (Richard Bellman)
- *Research:*
  - How to compute an optimal policy?
  - MDPs as models in economics etc.
  - *Applications:*
    Inventory management, maintenance management, routing in networks, ...

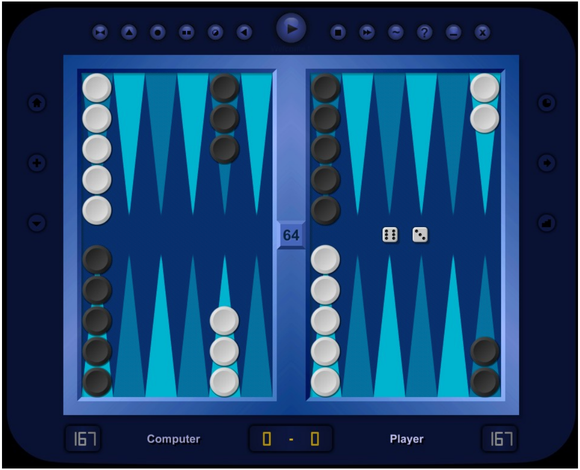
- *1940s:* dynamic programming for optimization problems (Richard Bellman)
- *1950s: stochastic* dynamic programming $\rightarrow$ *MDPs* (Richard Bellman)
- *Research:*
  - How to compute an optimal policy?
  - MDPs as models in economics etc.
  - *Applications:*
    Inventory management, maintenance management, routing in networks, ...
- *1980s: Artificial intelligence* discovers MDPs as models for learning with delayed feedback $\rightarrow$ *Reinforcement Learning* (MDP as model for the unknown environment)

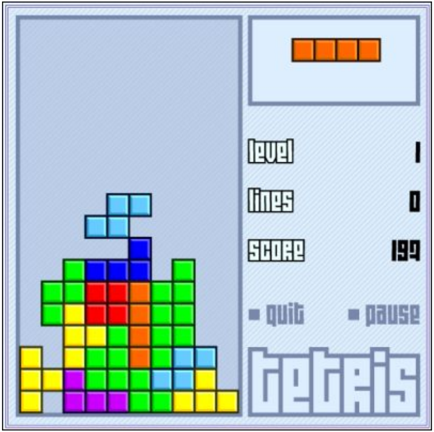
MDPs: More examples I



MDPs: More examples II




64
167
Computer
0 - 0
Player
167


level
1
lines
0
SCORE
199
quit
pause
tetris

Definition

Markov decision process (MDP) $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, s_1, p, r \rangle$:

$$
\begin{array}{lll}
\mathcal{S} & \ldots & \text{state space} \\
\mathcal{A} & \ldots & \text{a set of actions available in each state}
\end{array}
$$

- Start in initial state s_1.
- When choosing action a in state s:
  - ▷ random reward with mean $r(s, a)$ in $[0, 1]$,
  - ▷ transition to state s' with probability $p(s'|s, a)$.

Definition

A *(stationary) policy* on an MDP $\mathcal{M}$ is a mapping $\pi : \mathcal{S} \to \mathcal{A}$.

Definition

The *average reward* of a policy is

$$\rho(\mathcal{M}, \pi) := \lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} r\big(s_t, \pi(s_t)\big),$$

where s_t is a random variable for the state at step t.

We are interested in the optimal policy π^* giving optimal average reward $\rho^* := \max_\pi(\mathcal{M}, \pi)$.

Any (stationary) policy induces a *Markov chain* on the MDP.

Definition

A *Markov chain* on state space $\mathcal{S}$ is a sequence of random variables $S_t \in \mathcal{S}$ such that:

- (Markov property) The probability of being in state s at time t depends only on the state at time $t-1$, that is,

$$\mathbb{P}\{S_t = s | S_1 = s_1, \ldots, S_{t-1} = s_{t-1}\} = \mathbb{P}\{S_t = s | S_{t-1} = s_{t-1}\}$$

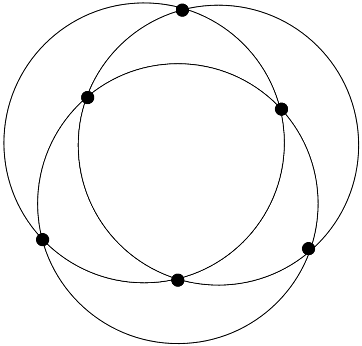
- (Time Homogeneity) The transition probability from state s to state s' does not depend on the time step, that is,

$$\mathbb{P}\{S_t = s | S_{t-1} = s'\} = \mathbb{P}\{S_{t'} = s | S_{t'-1} = s'\}$$

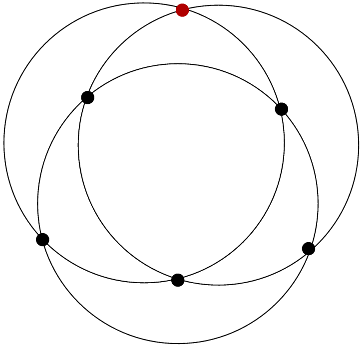
Consequently, a Markov chain is defined by

- the state space $\mathcal{S}$,
- an initial state $s_1 \in \mathcal{S}$, or more generally an initial distribution over $\mathcal{S}$,
- a quadratic transition matrix P such that $P_{s,s'} = \mathbb{P}\{S_t = s' | S_{t-1} = s\}$ is the probability of a transition to state s' when in state s.

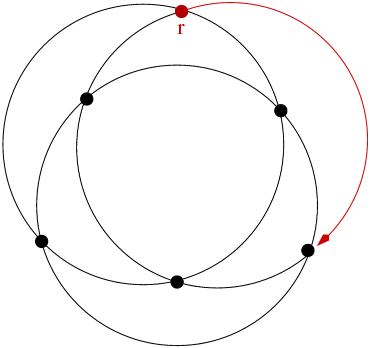
Random walk on a graph



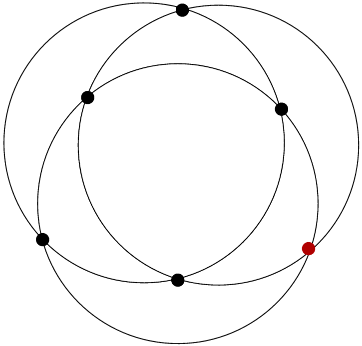
Random walk on a graph



Random walk on a graph



Random walk on a graph



Given an (irreducible, aperiodic) Markov chain there is a unique stationary distribution μ over the state space $\mathcal{S}$, such that (independent of the initial state) the t-step probabilities approach μ for $t \to \infty$. That is,

$$\mu(s) = \lim_{t \to \infty} \frac{\text{number of visits in } s}{t}.$$

Given an (irreducible, aperiodic) Markov chain there is a unique stationary distribution μ over the state space $\mathcal{S}$, such that (independent of the initial state) the t-step probabilities approach μ for $t \to \infty$. That is,

$$\mu(s) = \lim_{t\to\infty} \frac{\text{number of visits in } s}{t}.$$

Thus, if a policy π induces a Markov chain with stationary distribution μ_π, we can write the average reward as

$$\rho(\mathcal{M}, \pi) := \lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} r\big(s_t, \pi(s_t)\big) = \sum_{s\in S} \mu_\pi(s) \cdot r(s, \pi(s))$$

The stationary distribution μ of a Markov chain with transition matrix P can be computed by solving $\mu P = \mu$.

$\rightsquigarrow$ can compute the optimal policy as follows:

- For each policy π:
  - ▷ Compute the stationary distribution μ_π and the average reward ρ_π.
- Return policy with maximal ρ_π.

The stationary distribution μ of a Markov chain with transition matrix P can be computed by solving $\mu P = \mu$.

$\rightsquigarrow$ can compute the optimal policy as follows:

- For each policy π:
  - ▷ Compute the stationary distribution μ_π and the average reward ρ_π.
- Return policy with maximal ρ_π.

Problem: The number of policies $\pi : \mathcal{S} \to \mathcal{A}$ is A^S, where $A := \mathcal{A}$ and $S := \mathcal{S}$.

A better way to compute the optimal policy in a known MDP:

Value iteration

- Set $u_0(s) := 0$ for all states $s \in \mathcal{S}$.
- For $i > 0$ and all $s \in \mathcal{S}$ set the iterated state values to be

$$u_{i+1}(s) \quad := \quad \max_{a \in A} \left\{ r(s,a) \; + \sum_{s' \in \mathcal{S}} p(s'|s,a) u_i(s') \right\}.$$

Convergence (if there is non-periodic optimal policy)

For $i \to \infty$:

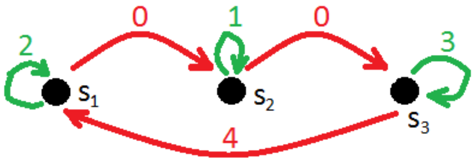
- The vector $(\mathbf{u}_{i+1} - \mathbf{u}_i)$ converges to $\mathbf{1}\rho^*$.
- The $\arg\max$-actions converge to an optimal policy.

Convergence (if there is non-periodic optimal policy)

For $i \to \infty$:

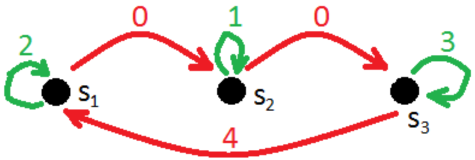
- The vector $(\mathbf{u}_{i+1} - \mathbf{u}_i)$ converges to $\mathbf{1}\rho^*$.
- The $\arg\max$-actions converge to an optimal policy.
- The quality of the greedy policy of the current iteration is measured by

$$\max_s \left\{u_{i+1}(s) - u_i(s)\right\} - \min_s \left\{u_{i+1}(s) - u_i(s)\right\}.$$



$$u_{i+1}(s) := \max_{a \in A} \left\{ r(s,a) + \sum_{s' \in \mathcal{S}} p(s'|s,a) u_i(s') \right\}.$$

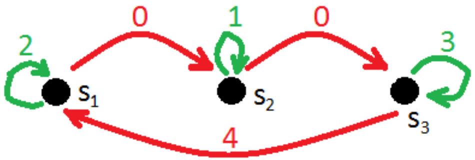
$u_0(s_1) = 0$ $\qquad$ $u_0(s_2) = 0$ $\qquad$ $u_0(s_3) = 0$ (initialization)



$$u_{i+1}(s) := \max_{a \in A} \left\{ r(s,a) + \sum_{s' \in \mathcal{S}} p(s'|s,a) u_i(s') \right\}.$$

$u_0(s_1) = 0$ $\quad u_0(s_2) = 0$ $\quad u_0(s_3) = 0$ (initialization)

$u_1(s_1) = 2$ $\quad u_1(s_2) = 1$ $\quad u_1(s_3) = 4$

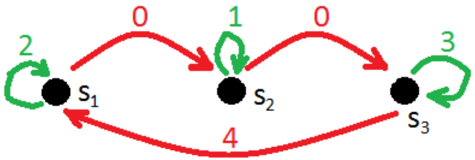


$$u_{i+1}(s) := \max_{a \in A} \left\{ r(s,a) + \sum_{s' \in \mathcal{S}} p(s'|s,a) u_i(s') \right\}.$$

$u_0(s_1) = 0$ $u_0(s_2) = 0$ $u_0(s_3) = 0$ (initialization)

$u_1(s_1) = 2$ $u_1(s_2) = 1$ $u_1(s_3) = 4$

$u_2(s_1) = \max\{2 + 2, 0 + 1\}$

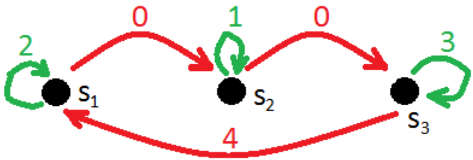


$$u_{i+1}(s) := \max_{a \in A} \left\{ r(s,a) + \sum_{s' \in \mathcal{S}} p(s'|s,a) u_i(s') \right\}.$$

$u_0(s_1) = 0$ $u_0(s_2) = 0$ $u_0(s_3) = 0$ (initialization)

$u_1(s_1) = 2$ $u_1(s_2) = 1$ $u_1(s_3) = 4$

$u_2(s_1) = 4$ $u_2(s_2) =$

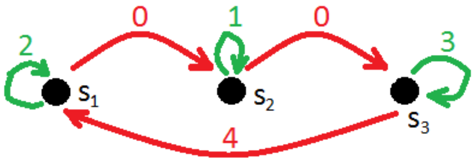


$$u_{i+1}(s) := \max_{a \in A} \left\{ r(s,a) + \sum_{s' \in \mathcal{S}} p(s'|s,a) u_i(s') \right\}.$$

$u_0(s_1) = 0$ $\quad$ $u_0(s_2) = 0$ $\quad$ $u_0(s_3) = 0$ (initialization)

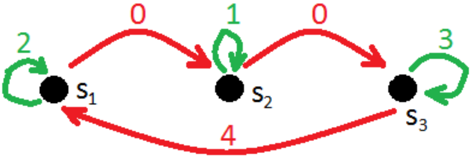
$u_1(s_1) = 2$ $\quad$ $u_1(s_2) = 1$ $\quad$ $u_1(s_3) = 4$

$u_2(s_1) = 4$ $\quad$ $u_2(s_2) = \max\{1 + 1, 0 + 4\}$



$$u_{i+1}(s) := \max_{a \in A} \left\{ r(s,a) + \sum_{s' \in \mathcal{S}} p(s'|s,a) u_i(s') \right\}.$$

$u_0(s_1) = 0$ $u_0(s_2) = 0$ $u_0(s_3) = 0$ (initialization)
$u_1(s_1) = 2$ $u_1(s_2) = 1$ $u_1(s_3) = 4$
$u_2(s_1) = 4$ $u_2(s_2) = 4^*$ $u_2(s_3) =$

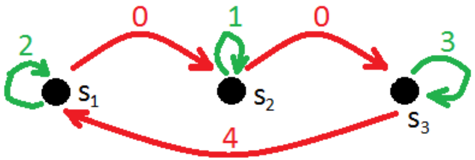


$$u_{i+1}(s) := \max_{a \in A} \left\{ r(s,a) + \sum_{s' \in \mathcal{S}} p(s'|s,a) u_i(s') \right\}.$$

$u_0(s_1) = 0$ $u_0(s_2) = 0$ $u_0(s_3) = 0$ (initialization)

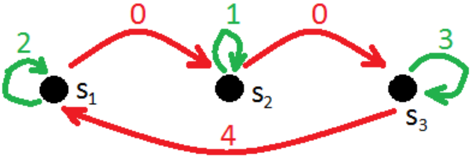
$u_1(s_1) = 2$ $u_1(s_2) = 1$ $u_1(s_3) = 4$

$u_2(s_1) = 4$ $u_2(s_2) = 4^*$ $u_2(s_3) = \max\{3 + 4, 4 + 2\}$



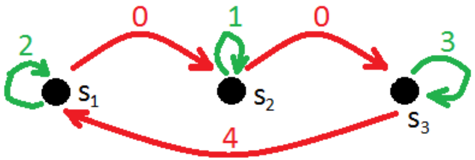
$$u_{i+1}(s) := \max_{a \in A} \left\{ r(s,a) + \sum_{s' \in \mathcal{S}} p(s'|s,a) u_i(s') \right\}.$$

$u_0(s_1) = 0$	$u_0(s_2) = 0$	$u_0(s_3) = 0$ (initialization)
$u_1(s_1) = 2$	$u_1(s_2) = 1$	$u_1(s_3) = 4$
$u_2(s_1) = 4$	$u_2(s_2) = 4^*$	$u_2(s_3) = 7^*$



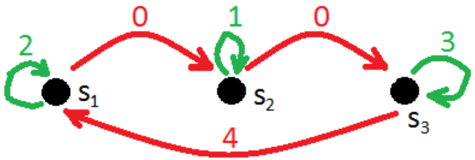
$$u_{i+1}(s) := \max_{a \in A} \left\{ r(s,a) + \sum_{s' \in \mathcal{S}} p(s'|s,a) u_i(s') \right\}.$$

$u_0(s_1) = 0$ $\quad$ $u_0(s_2) = 0$ $\quad$ $u_0(s_3) = 0$ (initialization)
$u_1(s_1) = 2$ $\quad$ $u_1(s_2) = 1$ $\quad$ $u_1(s_3) = 4$
$u_2(s_1) = 4$ $\quad$ $u_2(s_2) = 4^*$ $\quad$ $u_2(s_3) = 7^*$
$u_3(s_1) = \max\{2 + 4, 0 + 4\}$



$$u_{i+1}(s) := \max_{a \in A} \left\{ r(s,a) + \sum_{s' \in \mathcal{S}} p(s'|s,a) u_i(s') \right\}.$$

$u_0(s_1) = 0$	$u_0(s_2) = 0$	$u_0(s_3) = 0$ (initialization)
$u_1(s_1) = 2$	$u_1(s_2) = 1$	$u_1(s_3) = 4$
$u_2(s_1) = 4$	$u_2(s_2) = 4^*$	$u_2(s_3) = 7^*$
$u_3(s_1) = 6$	$u_3(s_2) =$	



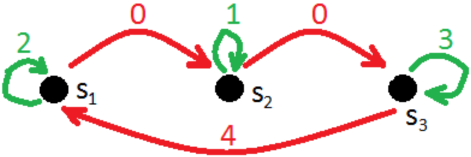
$$u_{i+1}(s) := \max_{a \in A} \left\{ r(s,a) + \sum_{s' \in \mathcal{S}} p(s'|s,a) u_i(s') \right\}.$$

$u_0(s_1) = 0$ | $u_0(s_2) = 0$ | $u_0(s_3) = 0$ (initialization)

$u_1(s_1) = 2$ | $u_1(s_2) = 1$ | $u_1(s_3) = 4$

$u_2(s_1) = 4$ | $u_2(s_2) = 4^*$ | $u_2(s_3) = 7^*$

$u_3(s_1) = 6$ | $u_3(s_2) = \max\{1 + 4, 0 + 7\}$



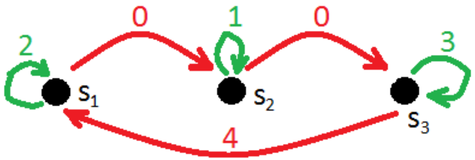
$$u_{i+1}(s) := \max_{a\in A}\left\{ r(s,a) + \sum_{s'\in\mathcal{S}} p(s'|s,a)u_i(s') \right\}.$$

$u_0(s_1) = 0$ $\quad$ $u_0(s_2) = 0$ $\quad$ $u_0(s_3) = 0$ (initialization)

$u_1(s_1) = 2$ $\quad$ $u_1(s_2) = 1$ $\quad$ $u_1(s_3) = 4$

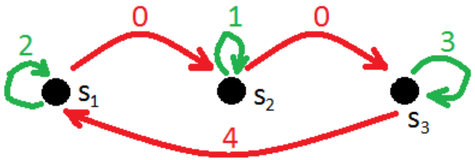
$u_2(s_1) = 4$ $\quad$ $u_2(s_2) = 4^*$ $\quad$ $u_2(s_3) = 7^*$

$u_3(s_1) = 6$ $\quad$ $u_3(s_2) = 7^*$ $\quad$ $u_3(s_3) =$



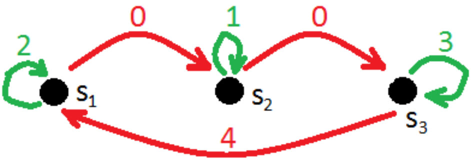
$$u_{i+1}(s) := \max_{a \in A} \left\{ r(s,a) + \sum_{s' \in \mathcal{S}} p(s'|s,a) u_i(s') \right\}.$$

$u_0(s_1) = 0$	$u_0(s_2) = 0$	$u_0(s_3) = 0$ (initialization)
$u_1(s_1) = 2$	$u_1(s_2) = 1$	$u_1(s_3) = 4$
$u_2(s_1) = 4$	$u_2(s_2) = 4^*$	$u_2(s_3) = 7^*$
$u_3(s_1) = 6$	$u_3(s_2) = 7^*$	$u_3(s_3) = \max\{3+7, 4+4\}$



$$u_{i+1}(s) := \max_{a \in A} \left\{ r(s,a) + \sum_{s' \in \mathcal{S}} p(s'|s,a) u_i(s') \right\}.$$

$u_0(s_1) = 0$	$u_0(s_2) = 0$	$u_0(s_3) = 0$ (initialization)
$u_1(s_1) = 2$	$u_1(s_2) = 1$	$u_1(s_3) = 4$
$u_2(s_1) = 4$	$u_2(s_2) = 4^*$	$u_2(s_3) = 7^*$
$u_3(s_1) = 6$	$u_3(s_2) = 7^*$	$u_3(s_3) = 10^*$



$$u_{i+1}(s) := \max_{a \in A} \left\{ r(s,a) + \sum_{s' \in \mathcal{S}} p(s'|s,a) u_i(s') \right\}.$$

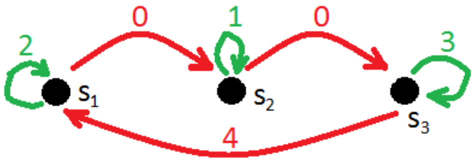
$u_0(s_1) = 0$ | $u_0(s_2) = 0$ | $u_0(s_3) = 0$ (initialization)

$u_1(s_1) = 2$ | $u_1(s_2) = 1$ | $u_1(s_3) = 4$

$u_2(s_1) = 4$ | $u_2(s_2) = 4^*$ | $u_2(s_3) = 7^*$

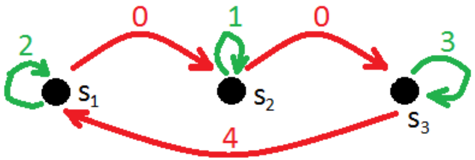
$u_3(s_1) = 6$ | $u_3(s_2) = 7^*$ | $u_3(s_3) = 10^*$

$u_4(s_1) =$ | $u_4(s_2) =$ | $u_4(s_3) =$



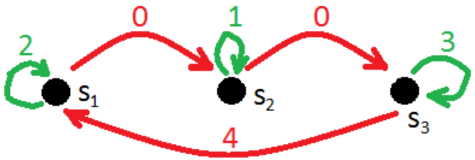
$$u_{i+1}(s) := \max_{a \in A} \left\{ r(s,a) + \sum_{s' \in \mathcal{S}} p(s'|s,a) u_i(s') \right\}.$$

$u_0(s_1) = 0$	$u_0(s_2) = 0$	$u_0(s_3) = 0$ (initialization)
$u_1(s_1) = 2$	$u_1(s_2) = 1$	$u_1(s_3) = 4$
$u_2(s_1) = 4$	$u_2(s_2) = 4^*$	$u_2(s_3) = 7^*$
$u_3(s_1) = 6$	$u_3(s_2) = 7^*$	$u_3(s_3) = 10^*$
$u_4(s_1) = 8$	$u_4(s_2) =$	$u_4(s_3) =$



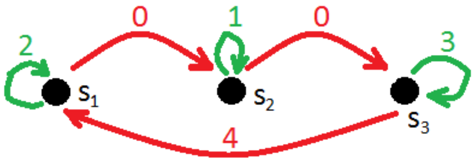
$$u_{i+1}(s) := \max_{a \in A} \left\{ r(s,a) + \sum_{s' \in \mathcal{S}} p(s'|s,a) u_i(s') \right\}.$$

$u_0(s_1) = 0$	$u_0(s_2) = 0$	$u_0(s_3) = 0$ (initialization)
$u_1(s_1) = 2$	$u_1(s_2) = 1$	$u_1(s_3) = 4$
$u_2(s_1) = 4$	$u_2(s_2) = 4^*$	$u_2(s_3) = 7^*$
$u_3(s_1) = 6$	$u_3(s_2) = 7^*$	$u_3(s_3) = 10^*$
$u_4(s_1) = 8$	$u_4(s_2) = 10^*$	$u_4(s_3) =$



$$u_{i+1}(s) := \max_{a \in A} \left\{ r(s,a) + \sum_{s' \in \mathcal{S}} p(s'|s,a) u_i(s') \right\}.$$

$u_0(s_1) = 0$	$u_0(s_2) = 0$	$u_0(s_3) = 0$ (initialization)
$u_1(s_1) = 2$	$u_1(s_2) = 1$	$u_1(s_3) = 4$
$u_2(s_1) = 4$	$u_2(s_2) = 4^*$	$u_2(s_3) = 7^*$
$u_3(s_1) = 6$	$u_3(s_2) = 7^*$	$u_3(s_3) = 10^*$
$u_4(s_1) = 8$	$u_4(s_2) = 10^*$	$u_4(s_3) = 13^*$



$$u_{i+1}(s) := \max_{a \in A} \left\{ r(s,a) + \sum_{s' \in \mathcal{S}} p(s'|s,a) u_i(s') \right\}.$$

$u_0(s_1) = 0$ $u_0(s_2) = 0$ $u_0(s_3) = 0$ (initialization)

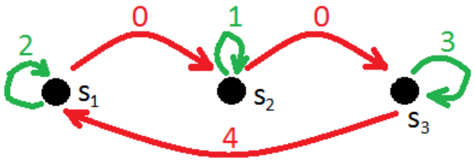
$u_1(s_1) = 2$ $u_1(s_2) = 1$ $u_1(s_3) = 4$

$u_2(s_1) = 4$ $u_2(s_2) = 4^*$ $u_2(s_3) = 7^*$

$u_3(s_1) = 6$ $u_3(s_2) = 7^*$ $u_3(s_3) = 10^*$

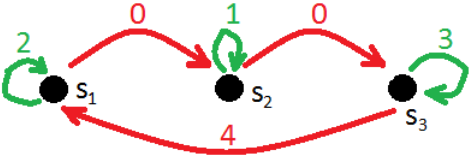
$u_4(s_1) = 8$ $u_4(s_2) = 10^*$ $u_4(s_3) = 13^*$

$u_5(s_1) = 10^{(*)}$ $u_5(s_2) = 13^*$ $u_5(s_3) = 16^*$



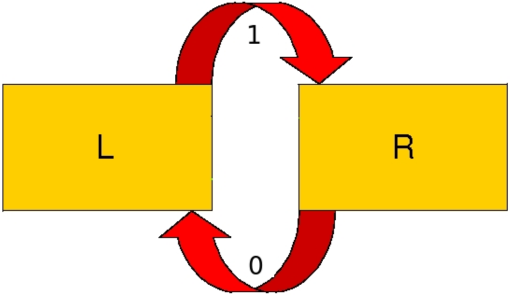
$$u_{i+1}(s) := \max_{a \in A} \left\{ r(s,a) + \sum_{s' \in \mathcal{S}} p(s'|s,a) u_i(s') \right\}.$$

$u_0(s_1) = 0$	$u_0(s_2) = 0$	$u_0(s_3) = 0$ (initialization)
$u_1(s_1) = 2$	$u_1(s_2) = 1$	$u_1(s_3) = 4$
$u_2(s_1) = 4$	$u_2(s_2) = 4^*$	$u_2(s_3) = 7^*$
$u_3(s_1) = 6$	$u_3(s_2) = 7^*$	$u_3(s_3) = 10^*$
$u_4(s_1) = 8$	$u_4(s_2) = 10^*$	$u_4(s_3) = 13^*$
$u_5(s_1) = 10^{(*)}$	$u_5(s_2) = 13^*$	$u_5(s_3) = 16^*$
$u_6(s_1) = 13^*$	$u_6(s_2) = 16^*$	$u_6(s_3) = 19^*$



$$u_{i+1}(s) := \max_{a \in A} \left\{ r(s,a) + \sum_{s' \in \mathcal{S}} p(s'|s,a) u_i(s') \right\}.$$

$u_0(s_1) = 0$	$u_0(s_2) = 0$	$u_0(s_3) = 0$ (initialization)
$u_1(s_1) = 2$	$u_1(s_2) = 1$	$u_1(s_3) = 4$
$u_2(s_1) = 4$	$u_2(s_2) = 4^*$	$u_2(s_3) = 7^*$
$u_3(s_1) = 6$	$u_3(s_2) = 7^*$	$u_3(s_3) = 10^*$
$u_4(s_1) = 8$	$u_4(s_2) = 10^*$	$u_4(s_3) = 13^*$
$u_5(s_1) = 10^{(*)}$	$u_5(s_2) = 13^*$	$u_5(s_3) = 16^*$
$u_6(s_1) = 13^*$	$u_6(s_2) = 16^*$	$u_6(s_3) = 19^*$
$u_7(s_1) = 16^*$	$u_7(s_2) = 19^*$	$u_7(s_3) = 22^*$



- Average reward $\rho = \frac{1}{2}$.
- Obviously, it's better to start in L.
- Can we quantify this?

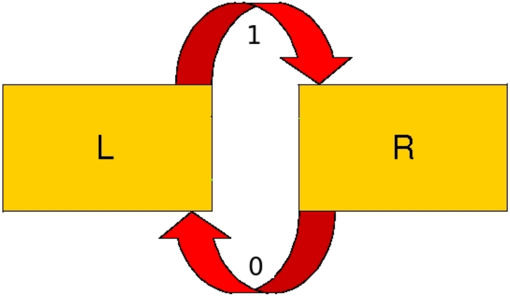
The Poisson equation relates the average reward $\rho(\pi)$ of a policy π to the individual rewards $r(s, \pi(s))$.

Poisson equation

$$\rho(\pi) - r(s, \pi(s)) = \sum_{s'} p(s'|s, \pi(s)) \cdot \lambda_\pi(s') - \lambda_\pi(s),$$

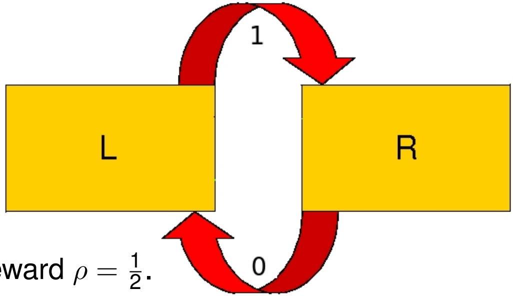
where $\lambda_\pi(s)$ is the bias of state s.

Intuitively, the bias indicates how much you gain/lose in accumulated rewards w.r.t. average reward when starting in state s.

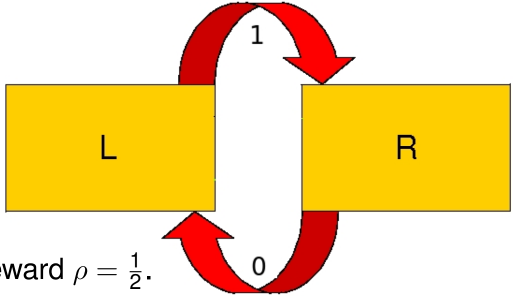


- Average reward $\rho = \frac{1}{2}$.
- Poisson equation:

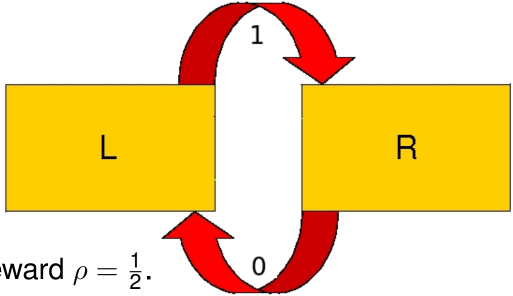
$$\begin{aligned} \rho - r(L) &= \lambda(R) - \lambda(L) \\ \rho - r(R) &= \lambda(L) - \lambda(R) \end{aligned}$$



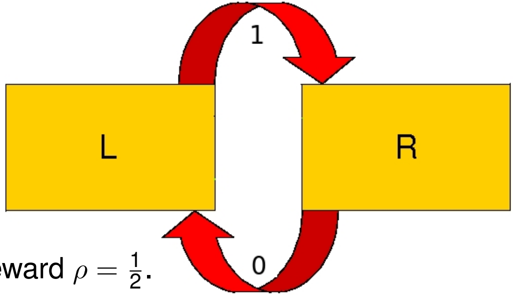
- Average reward $\rho = \frac{1}{2}$.
- Bias $\lambda(L) = \frac{1}{4}$, $\lambda(R) = -\frac{1}{4}$.
- **Interpretation:** Accumulated reward after $t = 1, 2, \ldots$ steps ...
  - ... when starting in L: $1, 1, 2, 2, 3, 3, 4, 4, \ldots$
  - ... when starting in R: $0, 1, 1, 2, 2, 3, 3, 4, \ldots$



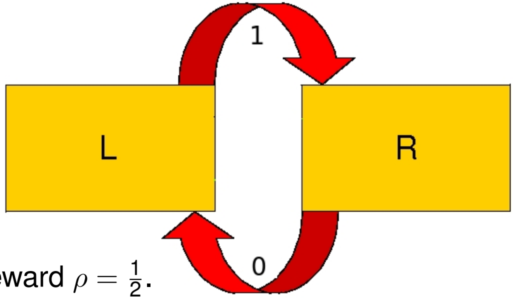
- Average reward $\rho = \frac{1}{2}$.
- Bias $\lambda(L) = \frac{1}{4}$, $\lambda(R) = -\frac{1}{4}$.
- **Interpretation:** Accumulated reward after $t = 1, 2, \dots$ steps ...
  - ... when starting in L: $1, 1, 2, 2, 3, 3, 4, 4, \dots$
  - ... when starting in R: $0, 1, 1, 2, 2, 3, 3, 4, \dots$
  - accum. average reward: $\frac{1}{2}, 1, \frac{3}{2}, 2, \frac{5}{2}, 3, \frac{7}{2}, 4, \dots$



- Average reward $\rho = \frac{1}{2}$.
- Bias $\lambda(L) = \frac{1}{4}$, $\lambda(R) = -\frac{1}{4}$.
- **Interpretation:** Accumulated reward after $t = 1, 2, \ldots$ steps ...
  - ... when starting in L: $1, 1, 2, 2, 3, 3, 4, 4, \ldots$
  - accum. average reward: $\frac{1}{2}, 1, \frac{3}{2}, 2, \frac{5}{2}, 3, \frac{7}{2}, 4, \ldots$
  - $\rightsquigarrow$ diff. sequence for L: $\frac{1}{2}, 0, \frac{1}{2}, 0, \frac{1}{2}, 0, \frac{1}{2}, 0, \ldots \to$ on avg. $\frac{1}{4}$



- Average reward $\rho = \frac{1}{2}$.
- Bias $\lambda(L) = \frac{1}{4}$, $\lambda(R) = -\frac{1}{4}$.
- **Interpretation:** Accumulated reward after $t = 1, 2, \ldots$ steps ...
  - ... when starting in R: $0, 1, 1, 2, 2, 3, 3, 4, \ldots$
  - accum. average reward: $\frac{1}{2}, 1, \frac{3}{2}, 2, \frac{5}{2}, 3, \frac{7}{2}, 4, \ldots$
  - $\rightsquigarrow$ diff. sequence for R: $-\frac{1}{2}, 0, -\frac{1}{2}, 0, -\frac{1}{2}, 0, \ldots \to$ on avg. $-\frac{1}{4}$



- Average reward $\rho = \frac{1}{2}$.
- Bias $\lambda(L) = \frac{1}{4}$, $\lambda(R) = -\frac{1}{4}$.
- **Interpretation:** Accumulated reward after $t = 1, 2, \ldots$ steps ...
  - ... when starting in L: $1, 1, 2, 2, 3, 3, 4, 4, \ldots$
  - ... when starting in R: $0, 1, 1, 2, 2, 3, 3, 4, \ldots$
  - $\rightsquigarrow$ difference sequence: $1, 0, 1, 0, 1, 0, 1, 0, \ldots$
  - average difference $= \frac{1}{2} = \lambda(L) - \lambda(R)$ “bias span”

Definition

The *diameter* of an MDP is the maximal expected time it takes to reach one state from any other state (using an appropriate policy).

- Intuitively, the bias indicates how much you gain/lose in accumulated rewards w.r.t. average reward when starting in state s.
- If the rewards are bounded in $[0, 1]$, the bias span of the optimal policy is bounded by the diameter.

The Learner's Goal(s):

1. Find optimal policy $\pi^* = \arg\max_\pi \rho(\mathcal{M}, \pi)$.
2. Do this online, so that you don't lose too much w.r.t. $\rho^* := \rho(\mathcal{M}, \pi^*)$.

The Learner's Goal(s):

1. Find optimal policy $\pi^* = \arg\max_\pi \rho(\mathcal{M}, \pi)$.
2. Do this online, so that you don't lose too much w.r.t. $\rho^* := \rho(\mathcal{M}, \pi^*)$.

$\rightsquigarrow$ Maximize $\sum_{t=1}^{T} r_t$, where r_t is the random reward at step t.

The Learner's Goal(s):

1. Find optimal policy $\pi^* = \arg\max_\pi \rho(\mathcal{M}, \pi)$.
2. Do this online, so that you don't lose too much w.r.t. $\rho^* := \rho(\mathcal{M}, \pi^*)$.

$\rightsquigarrow$ Maximize $\sum_{t=1}^{T} r_t$, where r_t is the random reward at step t.
$\rightsquigarrow$ Minimize the *regret*:

Definition

The learner's *regret* after T steps is

$$T\rho^* - \sum_{t=1}^{T} r_t.$$

- Consider each policy as the arm of a bandit problem.
- Use a bandit algorithm to select a policy.
- Play the policy for sufficiently many steps.

- Consider each policy as the arm of a bandit problem.
- Use a bandit algorithm to select a policy.
- Play the policy for sufficiently many steps.

Problem: The number of policies $\pi : \mathcal{S} \to \mathcal{A}$ is A^S.

Estimates:

- **In bandit case:** estimates $\hat{r}(a)$ for reward of each arm a
- **For MDPs:** estimates for rewards and transition probabilities:

$$\hat{r}(s,a) := \frac{\text{total reward when playing } a \text{ in } s}{\text{number of visits in } s, a},$$

$$\hat{p}(s'|s,a) := \frac{\text{total number of transitions to } s' \text{ when playing } a \text{ in } s}{\text{number of visits in } s, a}.$$

Confidence intervals:

- **In bandit case:** confidence intervals for reward of each arm
- **For MDPs:** confidence intervals for rewards and transition probabilities

$\rightsquigarrow$ The set $\mathbb{M}$ of plausible MDPs given the estimates $\hat{r}$ and $\hat{p}$ is the set of all MDPs with rewards r' and transition probabilities p' such that

$$\begin{aligned}
\left|\hat{r}(s,a) - r'(s,a)\right| &\leq \mathrm{conf}_r(s,a) := \sqrt{\frac{3\log(2SAt/\delta)}{N(s,a)}}, \\
\left\|\hat{p}(\cdot|s,a) - p'(\cdot|s,a)\right\|_1 &\leq \mathrm{conf}_p(s,a) := \sqrt{\frac{12S\log(2At/\delta)}{N(s,a)}}.
\end{aligned}$$

Optimism:

- **In bandit case:** Choose arm with highest upper confidence bound.
- **For MDPs:** Choose plausible MDP $\tilde{\mathcal{M}} \in \mathbb{M}$ that promises highest average reward under optimal policy.

$\rightsquigarrow$ Choose optimistic MDP $\tilde{\mathcal{M}} \in \mathbb{M}$ and optimal policy $\tilde{\pi}$ such that

$$\rho(\tilde{\mathcal{M}}, \tilde{\pi}) = \max_{\pi, \mathcal{M} \in \mathbb{M}} \rho(\mathcal{M}, \pi).$$

Choose optimistic MDP $\tilde{\mathcal{M}} \in \mathbb{M}$ and optimal policy $\tilde{\pi}$ such that

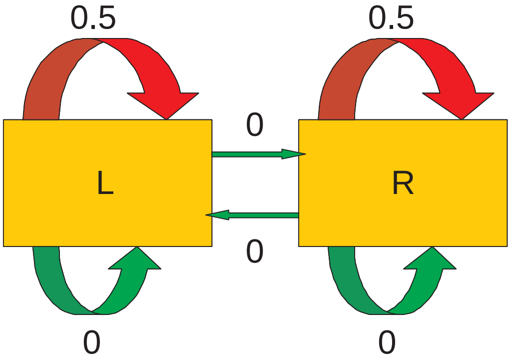
$$\rho(\tilde{\mathcal{M}}, \tilde{\pi}) = \max_{\pi, \mathcal{M} \in \mathbb{M}} \rho(\mathcal{M}, \pi).$$

- Set rewards $\tilde{r}$ to the upper confidence bounds.
- For the transition probabilities $\tilde{p}$ one can use an extension of value iteration. That is, for all states s set $u_0(s) := 0$ and

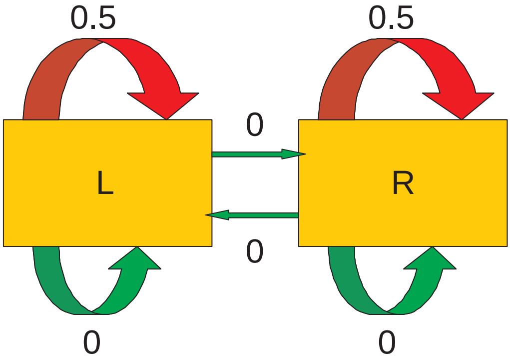
$$u_{i+1}(s) \quad := \quad \max_a \left\{ \tilde{r}(s,a) + \max_{p \in \mathcal{P}(s,a)} \left\{ \sum_{s'} p(s') u_i(s') \right\} \right\},$$

where $\mathcal{P}(s,a)$ is the set of all plausible transitions from s, a.

It's a bad idea to change the policy too often.



It's a bad idea to change the policy too often.



It depends on the bias how fast we approach the average reward of the chosen policy!

UCRL2 (Jaksch et al., 2010)

For episodes $k = 1, 2, \ldots$ do:

1. Maintain UCB-like confidence intervals for rewards and transition probabilities to define set of plausible MDPs $\mathbb{M}$.
2. Calculate optimal policy $\tilde{\pi}$ in optimistic model $\tilde{\mathcal{M}} \in \mathbb{M}$, i.e.

$$\rho(\tilde{\mathcal{M}}, \tilde{\pi}) = \max_{\pi, \mathcal{M} \in \mathbb{M}} \rho(\mathcal{M}, \pi).$$

3. Execute $\tilde{\pi}$ until the visits in some state-action pair have doubled.

We first consider the regret in a single episode k.

Since w.h.p. the true MDP $\mathcal{M}$ is in $\mathbb{M}$, we have for the policy $\tilde{\pi}_k$ chosen in episode k

$$\tilde{\rho}(\tilde{\mathcal{M}}, \tilde{\pi}_k) \ \geq \ \rho^* \ = \ \rho(\mathcal{M}, \pi^*) \geq \ \rho(\mathcal{M}, \tilde{\pi}_k).$$

We first consider the regret in a single episode k.

Since w.h.p. the true MDP $\mathcal{M}$ is in $\mathbb{M}$, we have for the policy $\tilde{\pi}_k$ chosen in episode k

$$\tilde{\rho}(\tilde{\mathcal{M}}, \tilde{\pi}_k) \geq \rho^* = \rho(\mathcal{M}, \pi^*) \geq \rho(\mathcal{M}, \tilde{\pi}_k).$$

Intuitively, the regret is upper bounded by the sum over the confidence intervals in each step

$$\sum_k \sum_{s,a} v_k(s,a) \cdot \mathrm{conf_k(s,a)},$$

where $v_k(s,a)$ are the visits of s, a in episode k.

We first consider the regret in a single episode k.

Since w.h.p. the true MDP $\mathcal{M}$ is in $\mathbb{M}$, we have for the policy $\tilde{\pi}_k$ chosen in episode k

$$\tilde{\rho}_k := \tilde{\rho}(\tilde{\mathcal{M}}, \tilde{\pi}_k) \geq \rho^* = \rho(\mathcal{M}, \pi^*) \geq \rho(\mathcal{M}, \tilde{\pi}_k).$$

Hence, the regret in episode k is bounded by

$$\sum_{t=t_k}^{t_{k+1}-1} (\rho^* - r_t) \leq \sum_{t=t_k}^{t_{k+1}-1} (\tilde{\rho}_k - r_t),$$

where t_k is the time step when episode k starts.

Hence, the regret in episode k is bounded by

$$\sum_{t=t_k}^{t_{k+1}-1} (\rho^* - r_t) \leq \sum_{t=t_k}^{t_{k+1}-1} (\tilde{\rho}_k - r_t).$$

Hence, the regret in episode k is bounded by

$$\sum_{t=t_k}^{t_{k+1}-1} (\rho^* - r_t) \leq \sum_{t=t_k}^{t_{k+1}-1} (\tilde{\rho}_k - r_t)\,.$$

Ignoring the random fluctuation of the rewards, we can write

$$\sum_{t=t_k}^{t_{k+1}-1} r_t \approx \sum_{s,a} v_k(s,a)\, r(s,a)\,.$$

Hence, the regret in episode k is bounded by

$$\sum_{t=t_k}^{t_{k+1}-1} (\rho^* - r_t) \le \sum_{t=t_k}^{t_{k+1}-1} (\tilde{\rho}_k - r_t) .$$

Ignoring the random fluctuation of the rewards, we can write

$$\sum_{t=t_k}^{t_{k+1}-1} r_t \approx \sum_{s,a} v_k(s,a)\, r(s,a) .$$

Hence, the regret in episode k is bounded by

$$\sum_{s,a} v_k(s,a) \big(\tilde{\rho}_k - r(s,a)\big).$$

Hence, the regret in episode k is bounded by

$$\sum_{s,a} v_k(s,a)\big(\tilde{\rho}_k - r(s,a)\big) = \sum_{s,a} v_k(s,a)\big(\tilde{\rho}_k - \tilde{r}_k(s,a)\big) + \sum_{s,a} v_k(s,a)\big(\tilde{r}_k(s,a) - r(s,a)\big)$$

Hence, the regret in episode k is bounded by

$$\sum_{s,a} v_k(s,a)\big(\tilde{\rho}_k - r(s,a)\big) = \sum_{s,a} v_k(s,a)\big(\tilde{\rho}_k - \tilde{r}_k(s,a)\big) + \sum_{s,a} v_k(s,a)\big(\tilde{r}_k(s,a) - r(s,a)\big)$$

The second term is bounded by
$|\tilde{r}_k(s,a) - \hat{r}_k(s,a)| + |\hat{r}_k(s,a) - r(s,a)| \leq 2\mathrm{conf}_r(s,a)$.

Hence, the regret in episode k is bounded by

$$\sum_{s,a} v_k(s,a)\big(\tilde{\rho}_k - r(s,a)\big) \leq \sum_{s,a} v_k(s,a)\big(\tilde{\rho}_k - \tilde{r}_k(s,a)\big) + \sum_{s,a} v_k(s,a)\, 2\mathrm{conf}_r(s,a)$$

The second term is bounded by
$|\tilde{r}_k(s,a) - \hat{r}_k(s,a)| + |\hat{r}_k(s,a) - r(s,a)| \leq 2\mathrm{conf}_r(s,a)$.

Hence, the regret in episode k is bounded by

$$\sum_{s,a} v_k(s,a)\big(\tilde{\rho}_k - r(s,a)\big) \quad \leq \quad \sum_{s,a} v_k(s,a)\big(\tilde{\rho}_k - \tilde{r}_k(s,a)\big) + \sum_{s,a} v_k(s,a)\, 2\mathrm{conf}_r(s,a)$$

For the first term we use the Poisson equation

$$\tilde{\rho}(\tilde{\pi}_k) - \tilde{r}_k(s, \tilde{\pi}_k(s)) = \sum_{s'} \tilde{p}(s'|s, \tilde{\pi}_k(s)) \cdot \tilde{\lambda}_{\tilde{\pi}_k}(s') - \tilde{\lambda}_{\tilde{\pi}_k}(s).$$

Hence, the regret in episode k is bounded by

$$\sum_{s,a} v_k(s,a)\big(\tilde{\rho}_k - r(s,a)\big) \quad \leq \quad \sum_{s,a} v_k(s,a)\big(\tilde{\rho}_k - \tilde{r}_k(s,a)\big) + \sum_{s,a} v_k(s,a)\, 2\mathrm{conf}_r(s,a)$$

For the first term we use the Poisson equation

$$\tilde{\rho}(\tilde{\pi}_k) - \tilde{r}_k(s,\tilde{\pi}_k(s)) = \sum_{s'} \tilde{p}(s'|s,\tilde{\pi}_k(s)) \cdot \tilde{\lambda}_{\tilde{\pi}_k}(s') - \tilde{\lambda}_{\tilde{\pi}_k}(s).$$

(Note that $v_k(s,a) = 0$ for $a \neq \tilde{\pi}_k(s)$.)

Lemma

For any two states s, s′,

$$\tilde{\lambda}_{\tilde{\pi}_k}(s) - \tilde{\lambda}_{\tilde{\pi}_k}(s') \leq D,$$

where D is the diameter in the true MDP.

Proof sketch: Assume that $\tilde{\lambda}_{\tilde{\pi}_k}(s) - \tilde{\lambda}_{\tilde{\pi}_k}(s') > D$. Then one can define a nonstationary policy that goes from s' to s in at most D steps and employs the optimal policy from there. This gives higher reward than $\tilde{\pi}_k$, contradicting optimality of $\tilde{\pi}_k$.

Thus, we consider

$$\sum_{s,a} v_k(s,a)\big(\tilde{\rho}_k - \tilde{r}_k(s,a)\big)$$

$$= \sum_{s,a} v_k(s,a)\left(\sum_{s'} \tilde{p}(s'|s,\tilde{\pi}_k(s)) \cdot \tilde{\lambda}_{\tilde{\pi}_k}(s') - \tilde{\lambda}_{\tilde{\pi}_k}(s)\right)$$

Thus, we consider

$$\begin{aligned}
&\sum_{s,a} v_k(s,a)\big(\tilde{\rho}_k - \tilde{r}_k(s,a)\big) \\
&= \sum_{s,a} v_k(s,a)\Big(\sum_{s'} \tilde{p}(s'|s,\tilde{\pi}_k(s)) \cdot \tilde{\lambda}_{\tilde{\pi}_k}(s') - \tilde{\lambda}_{\tilde{\pi}_k}(s)\Big) \\
&= \mathbf{v}_k\big(\tilde{\mathbf{P}}_k - \mathbf{I}\big)\tilde{\lambda}_k
\end{aligned}$$

Thus, we consider

$$
\begin{aligned}
&\sum_{s,a} v_k(s,a)\big(\tilde{\rho}_k - \tilde{r}_k(s,a)\big) \\
&\quad = \sum_{s,a} v_k(s,a)\Bigg(\sum_{s'} \tilde{p}(s'|s,\tilde{\pi}_k(s)) \cdot \tilde{\lambda}_{\tilde{\pi}_k}(s') - \tilde{\lambda}_{\tilde{\pi}_k}(s)\Bigg) \\
&\quad = \mathbf{v}_k\big(\tilde{\mathbf{P}}_k - \mathbf{I}\big)\tilde{\lambda}_k \\
&\quad = \mathbf{v}_k\big(\tilde{\mathbf{P}}_k - \mathbf{P} + \mathbf{P} - \mathbf{I}\big)\tilde{\lambda}_k
\end{aligned}
$$

Thus, we consider

$$
\begin{aligned}
&\sum_{s,a} v_k(s,a)\big(\tilde\rho_k - \tilde r_k(s,a)\big) \\
&= \sum_{s,a} v_k(s,a)\left(\sum_{s'} \tilde p(s'|s,\tilde\pi_k(s)) \cdot \tilde\lambda_{\tilde\pi_k}(s') - \tilde\lambda_{\tilde\pi_k}(s)\right) \\
&= \mathbf{v}_k\big(\tilde{\mathbf{P}}_k - \mathbf{I}\big)\tilde{\boldsymbol\lambda}_k \\
&= \mathbf{v}_k\big(\tilde{\mathbf{P}}_k - \mathbf{P} + \mathbf{P} - \mathbf{I}\big)\tilde{\boldsymbol\lambda}_k \\
&= \mathbf{v}_k\big(\tilde{\mathbf{P}}_k - \mathbf{P}\big)\tilde{\boldsymbol\lambda}_k + \mathbf{v}_k\big(\mathbf{P} - \mathbf{I}\big)\tilde{\boldsymbol\lambda}_k.
\end{aligned}
$$

Thus, we consider

$$\begin{aligned}
&\sum_{s,a} v_k(s,a)\big(\tilde{\rho}_k - \tilde{r}_k(s,a)\big) \\
&= \sum_{s,a} v_k(s,a)\Big(\sum_{s'} \tilde{p}(s'|s,\tilde{\pi}_k(s)) \cdot \tilde{\lambda}_{\tilde{\pi}_k}(s') - \tilde{\lambda}_{\tilde{\pi}_k}(s)\Big) \\
&= \mathbf{v}_k\big(\tilde{\mathbf{P}}_k - \mathbf{P}\big)\tilde{\boldsymbol{\lambda}}_k + \mathbf{v}_k\big(\mathbf{P} - \mathbf{I}\big)\tilde{\boldsymbol{\lambda}}_k.
\end{aligned}$$

Thus, we consider

$$\begin{aligned}
&\sum_{s,a} v_k(s,a)\big(\tilde{\rho}_k - \tilde{r}_k(s,a)\big) \\
&= \sum_{s,a} v_k(s,a)\left(\sum_{s'} \tilde{p}(s'|s,\tilde{\pi}_k(s)) \cdot \tilde{\lambda}_{\tilde{\pi}_k}(s') - \tilde{\lambda}_{\tilde{\pi}_k}(s)\right) \\
&= \mathbf{v}_k(\tilde{\mathbf{P}}_k - \mathbf{P})\tilde{\boldsymbol{\lambda}}_k + \mathbf{v}_k(\mathbf{P} - \mathbf{I})\tilde{\boldsymbol{\lambda}}_k.
\end{aligned}$$

First term is bounded like

$$\begin{aligned}
\mathbf{v}_k(\tilde{\mathbf{P}}_k - \mathbf{P})\tilde{\boldsymbol{\lambda}}_k &\leq \left\|\mathbf{v}_k(\tilde{\mathbf{P}}_k - \mathbf{P})\right\|_1 \cdot \left\|\tilde{\boldsymbol{\lambda}}_k\right\|_\infty \\
&\leq 2\sum_{s,a} v_k(s,a)\,\mathrm{conf}_p(s,a)\, D.
\end{aligned}$$

Thus, we consider

$$
\begin{aligned}
&\sum_{s,a} v_k(s,a)\big(\tilde{\rho}_k - \tilde{r}_k(s,a)\big) \\
&\quad = \sum_{s,a} v_k(s,a)\left(\sum_{s'} \tilde{p}(s'|s,\tilde{\pi}_k(s)) \cdot \tilde{\lambda}_{\tilde{\pi}_k}(s') - \tilde{\lambda}_{\tilde{\pi}_k}(s)\right) \\
&\quad \leq 2\sum_{s,a} v_k(s,a)\,\mathrm{conf}_p(s,a) D + \mathbf{v}_k(\mathbf{P}-\mathbf{I})\tilde{\boldsymbol{\lambda}}_k.
\end{aligned}
$$

Second term can be rewritten as martingale difference sequence

$$
\begin{aligned}
\mathbf{v}_k(\mathbf{P}-\mathbf{I})\tilde{\boldsymbol{\lambda}}_k &= \sum_{t=t_k}^{t_{k+1}-1} \Big(p(\cdot|s_t,a)\tilde{\boldsymbol{\lambda}}_k - \tilde{\lambda}_k(s_t)\Big) \\
&= \sum_{t=t_k}^{t_{k+1}-1} \Big(p(\cdot|s_t,a)\tilde{\boldsymbol{\lambda}}_k - \tilde{\lambda}_k(s_{t+1})\Big) + \tilde{\lambda}_k(s_{t_{k+1}}) - \tilde{\lambda}_k(s_{t_k})
\end{aligned}
$$

Thus, we consider

$$\begin{aligned}
&\sum_{s,a} v_k(s,a)\big(\tilde{\rho}_k - \tilde{r}_k(s,a)\big) \\
&= \sum_{s,a} v_k(s,a)\Bigg(\sum_{s'} \tilde{p}(s'|s,\tilde{\pi}_k(s)) \cdot \tilde{\lambda}_{\tilde{\pi}_k}(s') - \tilde{\lambda}_{\tilde{\pi}_k}(s)\Bigg) \\
&= 2D\sum_{s,a} v_k(s,a)\,\mathrm{conf}_p(s,a) + \mathbf{v}_k\big(\mathbf{P}-\mathbf{I}\big)\tilde{\boldsymbol{\lambda}}_k.
\end{aligned}$$

Second term can be rewritten as martingale difference sequence

$$\mathbf{v}_k\big(\mathbf{P}-\mathbf{I}\big)\tilde{\boldsymbol{\lambda}}_k = \sum_{t=t_k}^{t_{k+1}-1}\Big(p(\cdot|s_t,a)\tilde{\boldsymbol{\lambda}}_k - \tilde{\lambda}_k(s_{t+1})\Big) + \tilde{\lambda}_k(s_{t_{k+1}}) - \tilde{\lambda}_k(s_{t_k})$$

and can be bounded by Azuma-Hoeffding inequality.

Theorem

Let $X_1, X_2, \ldots$ be a martingale difference sequence (i.e. $\mathbb{E}[X_i | X_1, \ldots, X_{i-1}] = 0$) with $|X_i| \leq c$ for all i.

Then for all $\varepsilon > 0$ and $n \in \mathbb{N}$,

$$\mathbb{P}\left\{\sum_{i=1}^{n} X_i \geq \varepsilon\right\} \leq \exp\left(-\frac{\varepsilon^2}{2nc^2}\right).$$

Since this last term is negligible compared to the main term, the regret in episode k is bounded by

$$\sum_{s,a} v_k(s,a)\big(\tilde{\rho}_k - r(s,a)\big)$$

$$\leq \quad \text{const} \cdot 2D \sum_{s,a} v_k(s,a)\, \text{conf}_p(s,a) + \text{const} \cdot 2 \sum_{s,a} v_k(s,a)\, \text{conf}_r(s,a).$$

Since this last term is negligible compared to the main term, the regret in episode k is bounded by

$$\sum_{s,a} v_k(s,a)\big(\tilde{\rho}_k - r(s,a)\big)$$

$$\leq \quad \text{const} \cdot 2D \sum_{s,a} v_k(s,a)\,\text{conf}_p(s,a) + \text{const} \cdot 2 \sum_{s,a} v_k(s,a)\,\text{conf}_r(s,a).$$

Summing over all episodes, the regret is bounded by

$$\begin{aligned}
&\sum_k \sum_{s,a} v_k(s,a)\big(\tilde{\rho}_k - r(s,a)\big) \\
&\leq \quad \text{const} \cdot D\sqrt{S\log(AT/\delta)} \sum_k \sum_{(s,a)} \frac{v_k(s,a)}{\sqrt{N_k(s,a)}} \\
&\leq \quad \text{const} \cdot D\sqrt{S\log(AT/\delta)}\sqrt{SAT} \\
&= \quad \text{const} \cdot DS\sqrt{AT\log(AT/\delta)}
\end{aligned}$$

Theorem (Jaksch et al., 2010)

In an MDP with S states, A actions, and diameter D with probability of at least $1-\delta$ the regret of UCRL2 after T steps is bounded by

$$34 \cdot DS\sqrt{AT \log\left(\frac{T}{\delta}\right)}.$$

Proof wrap-up:

$$\tilde{\rho}(\tilde{\pi}) \;\geq\; \rho^* \;\geq\; \rho(\tilde{\pi}),$$

so that the regret is upper bounded by the sum over the confidence intervals in each step

$$\sum_k \sum_{s,a} v_k(s,a) \cdot \mathrm{conf_k(s,a)} \;\leq\; const \cdot DS\sqrt{AT}.$$

Theorem (Jaksch et al., 2010)

In an MDP with S states, A actions, and diameter D with probability of at least $1-\delta$ the regret of UCRL2 after T steps is bounded by

$$34 \cdot DS\sqrt{AT \log\left(\frac{T}{\delta}\right)}.$$

Note: get sensible regret bound only for finite D!
(e.g., $D = \infty$ in the secretary problem!)

Theorem (Jaksch et al. 2010)

For any algorithm and any natural numbers T, S, $A > 1$, and $D \geq \log_A S$ there is an MDP $\mathcal{M}$ with S states, A actions, and diameter D, such that for any initial state $s \in \mathcal{S}$ the expected regret after T steps is

$$\Omega(\sqrt{DSAT}).$$

This is close to the upper bound, but there is a gap of $\sqrt{DS}$.

It is straightforward to obtain from the regret bound the following sample complexity bound.

Theorem (Jaksch et al., 2010)

With probability $1 - \delta$, *after*

$$T \geq 4 \cdot \frac{49^2 D^2 S^2 A}{\varepsilon^2} \log \left(\frac{49 DSA}{\delta \varepsilon} \right)$$

steps, the average per-step regret of UCRL2 is at most ε.

Theorem (Jaksch et al., 2010)

The expected regret of UCRL2 is

$$O\left(\frac{D^2 S^2 A \log(T)}{g}\right),$$

where g is the gap between the optimal average reward and the second largest average reward achievable in $\mathcal{M}$, that is,

$$g := \rho^*(\mathcal{M}) - \max_{\pi}\left\{\rho(\mathcal{M},\pi) : \rho(\mathcal{M},\pi) < \rho^*(\mathcal{M})\right\}.$$

- The logarithmic bound can be derived by considering the number L of suboptimal steps taken by UCRL2.
- As above, one can show an upper bound of $O(DS\sqrt{LA \log T})$ on the regret.
- As the loss in each suboptimal step is at least g, one has $gL = O(DS\sqrt{LA \log T})$, which gives

$$L = O\left(\frac{D^2 S^2 A \log T}{g^2}\right).$$

- A refined analysis of the regret in each suboptimal step improves the exponent of g and yields the claimed bound, as the regret of each suboptimal step is bounded by 1.

Outline

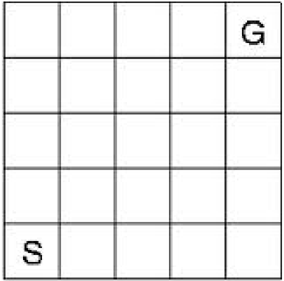
(Finite State) MDPs with additional similarity information:

Definition

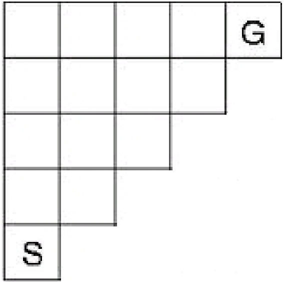
An ε-colored MDP is an MDP $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, s_0, p, r \rangle$ equipped with a coloring function $c : \mathcal{S} \times \mathcal{A} \to \mathcal{C}$ for a set of colors $\mathcal{C}$, such that:
If $c(s, a) = c(s', a')$ then

$$\begin{aligned} |r(s,a) - r(s',a')| &< \varepsilon, \\ \left\| p(\cdot|s,a) - p(\cdot|s',a') \right\|_1 &< \varepsilon. \end{aligned}$$

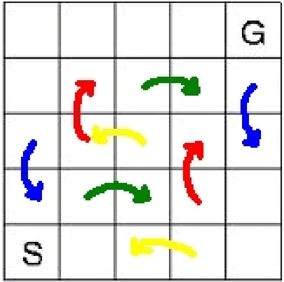
Idea: One sample of a state-action pair (s, a) gives information for all state-action pairs of the same color $c(s, a)$.



- No reduction of state space with ordinary aggregation.



- No reduction of state space with ordinary aggregation.
- Using homomorphisms (Ravindran& Barto, 2003):
  15 instead of 25 states, 4 actions



- No reduction of state space with ordinary aggregation.
- Using homomorphisms (Ravindran& Barto, 2003):
  15 instead of 25 states, 4 actions
- Colored MDP needs only as many colors as actions.
  Note: This does not necessarily reduce the MDP,
  but we can learn faster!

Colored UCRL2 (Ortner, Ryabko, Auer, & Munos 2012)

For episodes $k = 1, 2, \ldots$ do:

1. Maintain UCB-like confidence intervals $(+\varepsilon)$ for rewards and transition probabilities for each color to define set of plausible MDPs $\mathbb{M}$.
2. Calculate optimal policy $\tilde{\pi}$ in optimistic model $\tilde{\mathcal{M}} \in \mathbb{M}$, i.e.

$$\rho(\tilde{\mathcal{M}}, \tilde{\pi}) = \max_{\mathcal{M} \in \mathbb{M}, \pi} \rho(\mathcal{M}, \pi).$$

3. Execute $\tilde{\pi}$ until the visits for some color have doubled.

Theorem (Ortner, Ryabko, Auer, & Munos 2012)

In an ε-colored MDP with S states, C distinct colors, and diameter D, with probability of at least $1-\delta$ the regret of colored UCRL2 after T steps is bounded by

$$const \cdot D\sqrt{SCT \log\left(\tfrac{T}{\delta}\right)} + \varepsilon DT.$$

Proof Idea:

$$\tilde{\rho}(\tilde{\pi}) \;\geq\; \rho^* \;\geq\; \rho(\tilde{\pi}),$$

so that the regret is upper bounded by the sum over the confidence intervals in each step

$$\sum_k \sum_c v_k(c) \cdot \left(\mathrm{conf_k}(\mathrm{c}) + \varepsilon\right) \;\leq\; const \cdot D\sqrt{BCT} + \varepsilon DT.$$

Consider MDP with continuous state space where rewards and transition probabilities are Lipschitz or Hölder, that is,

Assumption

There are $L, \alpha > 0$ such that for any two states s, s' and all actions a,

$$\begin{aligned} |r(s,a) - r(s',a)| &\leq L|s-s'|^\alpha, \\ \left\|p(\cdot|s,a) - p(\cdot|s',a)\right\|_1 &\leq L|s-s'|^\alpha. \end{aligned}$$

Then close states behave similarly and if you discretize,
the situation is like in the colored MDP case.

For example, consider $\mathcal{S} = [0, 1]$.

- Then consider discretization $I_1 = \left[0, \frac{1}{n}\right]$, $I_2 = \left(\frac{1}{n}, \frac{2}{n}\right]$,..., $I_n = \left(\frac{n-1}{n}, 1\right]$.
- States within each interval have (by Lipschitz assumption) close rewards and transition probabilities.
- Discretization corresponds to coloring.

1. Original state space infinite.
2. $\rightsquigarrow$ The diameter is usually infinite.

To analyze the critical term in the regret

$$(t_{k+1} - t_k)\tilde{\rho}(\tilde{\pi}) - \sum_{t=t_k}^{t_{k+1}-1} \tilde{r}(s_t, \tilde{\pi}(s_t))$$

we

- use the Poisson equation for the optimistic MDP $\tilde{\mathcal{M}}$
- upper bound the bias $\tilde{\lambda}$ in $\tilde{\mathcal{M}}$ by the diameter D in true MDP $\mathcal{M}$.

Looking at the bound again ...

Theorem

In an MDP with S states, A actions, and diameter D with probability of at least $1 - \delta$ the regret of UCRL2 after T steps is bounded by

$$34 \cdot DS\sqrt{AT \log\left(\frac{T}{\delta}\right)}.$$

Proof Idea:

$$\tilde{\rho}(\tilde{\pi}) \;\geq\; \rho^* \;\geq\; \rho(\tilde{\pi}),$$

so that the regret is upper bounded by the sum over the confidence intervals in each step

$$\sum_k \sum_{s,a} v_k(s,a) \cdot \mathrm{conf_k(s,a)} \;\leq\; const \cdot DS\sqrt{AT}.$$

... there is an obvious question:
Shouldn't it be the bias span instead of the the diameter?

... there is an obvious question:
Shouldn't it be the bias span instead of the the diameter?

Yeah, but how do you relate the optimistic bias $\tilde{\lambda}_{\tilde{\pi}}$ to the real one?

... there is an obvious question:
Shouldn't it be the bias span instead of the the diameter?

Yeah, but how do you relate the optimistic bias $\tilde{\lambda}_{\tilde{\pi}}$ to the real one?

Well, you can cheat a bit:

- Look for optimistic model with bias bounded by the real bias.
- If you don't know the bias, try to guess it.
- That way you get regret bounds like for UCRL2 with the bias instead of the diameter.

... there is an obvious question:
Shouldn't it be the bias span instead of the the diameter?

Yeah, but how do you relate the optimistic bias $\tilde{\lambda}_{\tilde{\pi}}$ to the real one?

Well, you can cheat a bit:

- Look for optimistic model with bias bounded by the real bias.
- If you don't know the bias, try to guess it.
- That way you get regret bounds like for UCRL2 with the bias instead of the diameter.

Problem: UCRL2 finds optimistic model and optimal policy by extension of value iteration.

How about REGAL? We don't know.

UCCRL (Ortner & Ryabko 2012)

Input: Upper bound H on bias span of optimal policy,
Hölder parameters L, α, discretization parameter n

1. Discretize $[0, 1]$ into n intervals $I_1, \ldots, I_n$ of equal size.
2. For episodes $k = 1, 2, \ldots$ do:
   1. Maintain UCB-like confidence intervals $(+\varepsilon := Ln^{-\alpha})$ for rewards and transition probabilities of each interval I_j.
   2. Calculate optimal policy $\tilde{\pi}$ in optimistic model $\tilde{\mathcal{M}} \in \mathbb{M}$ under constraint that bias span of $\tilde{\pi}$ is upper bounded by H.

$$\rho(\tilde{\mathcal{M}}, \tilde{\pi}) = \max_{\pi, \mathcal{M} \in \mathbb{M}: H(M) \leq H} \rho(\mathcal{M}, \pi).$$

   3. Execute $\tilde{\pi}$ until the visits in some interval-action pair have doubled.

Theorem (Ortner & Ryabko 2012)

With probability $1 - \delta$ the regret of UCCRL after T steps is bounded by

$$const \cdot Hn\sqrt{AT \log\left(\frac{T}{\delta}\right)} + const \cdot HLn^{-\alpha}T.$$

Theorem (Ortner & Ryabko 2012)

With probability $1 - \delta$ the regret of UCCRL after T steps is bounded by

$$const \cdot Hn\sqrt{AT \log\left(\frac{T}{\delta}\right)} + const \cdot HLn^{-\alpha}T.$$

Choosing $n = T^{1/(2+2\alpha)}$ gives regret upper bounded by

$$const \cdot HL\sqrt{A \log\left(\frac{T}{\delta}\right)}T^{(2+\alpha)/(2+2\alpha)}.$$

Theorem (Ortner & Ryabko 2012)

With probability $1-\delta$ the regret of UCCRL after T steps is bounded by

$$const \cdot Hn\sqrt{AT \log\left(\frac{T}{\delta}\right)} + const \cdot HLn^{-\alpha} T.$$

Choosing $n = T^{1/(2+2\alpha)}$ gives regret upper bounded by

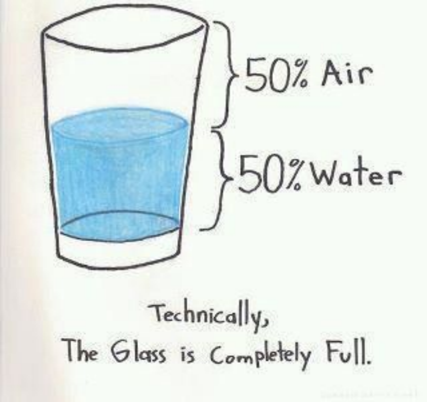
$$const \cdot HL\sqrt{A \log\left(\frac{T}{\delta}\right)}\, T^{(2+\alpha)/(2+2\alpha)}.$$

In particular, for Lipschitz MDPs the bound is $\tilde{O}(T^{3/4})$.

- Gap between algorithms for applications and algorithms with theoretical guarantees is still very large in general MDP setting.

- Gap between algorithms for applications and algorithms with theoretical guarantees is still very large in general MDP setting.
- wouldn't want to use UCRL2 in real-world application

- Gap between algorithms for applications and algorithms with theoretical guarantees is still very large in general MDP setting.
- wouldn't want to use UCRL2 in real-world application
- Still, optimism and confidence intervals work well to deal with exploration-exploitation problem.



Stopping Algorithm (Bruss 1984)

- Observe the first 37% of all options (but choose neither).
- Let $\hat{r}^*$ be the reward for the best option among the first 37%.

Stopping Algorithm (Bruss 1984)

- Observe the first 37% of all options (but choose neither).
- Let $\hat{r}^*$ be the reward for the best option among the first 37%.
- Choose the first option that has higher reward than $\hat{r}^*$.

Stopping Algorithm (Bruss 1984)

- Observe the first 37% of all options (but choose neither).
- Let $\hat{r}^*$ be the reward for the best option among the first 37%.
- Choose the first option that has higher reward than $\hat{r}^*$.

Theorem (Bruss 1984)

The stopping algorithm chooses the best option in 37% of all possible cases (permutation of the options).
This is also best possible.

- P. Auer, N. Cesa-Bianchi, and P. Fischer: Finite-time analysis of the multi-armed bandit problem. Mach. Learn. 47(2–3): 235–256, 2002.
- P. Auer, N. Cesa-Bianchi, Y. Freund, and R. Schapire: The non-stochastic multi-armed bandit problem. SIAM J. Computing 32(1): 48–77, 2002.
- T. Jaksch, R. Ortner, and P. Auer: Near-optimal regret bounds for reinforcement learning. J. Mach. Learn. Res. 11: 1563–1600, 2010.
- Shie Mannor and John N. Tsitsiklis: The sample complexity of exploration in the multi-armed bandit problem. J. Mach. Learn. Res. 5: 623–648, 2004.