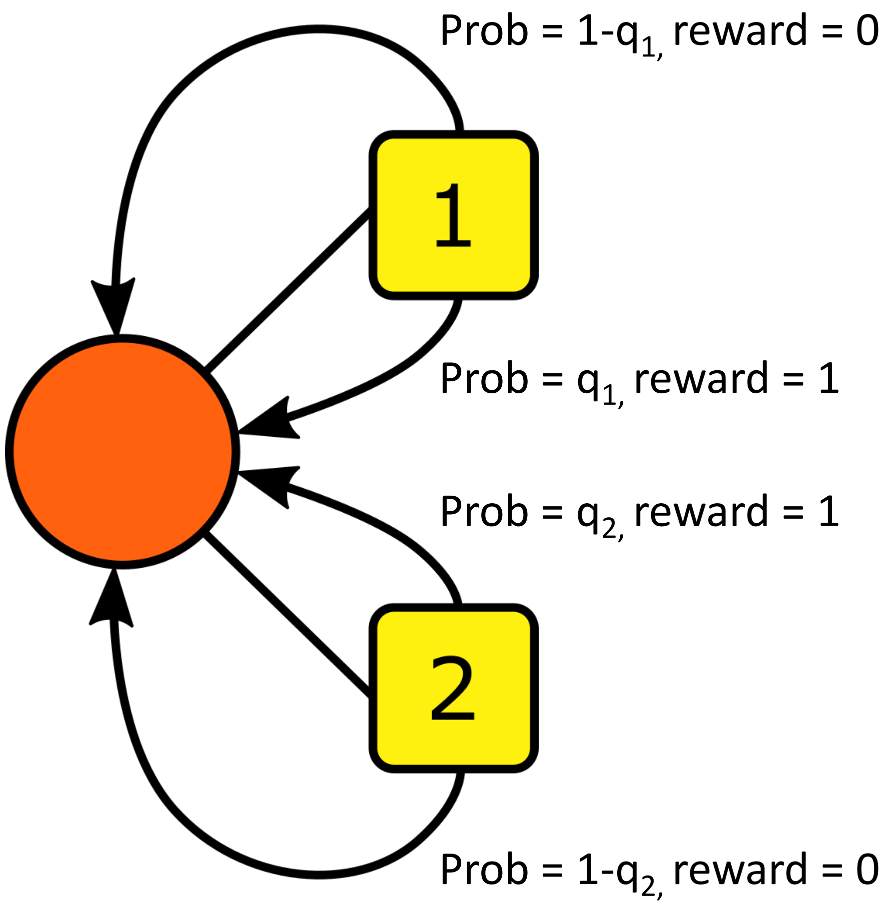
Two-armed Bernoulli bandits in the belief space

Andrea Mazzolini, HPC 2020

Bandit, hydrogen atom of RL



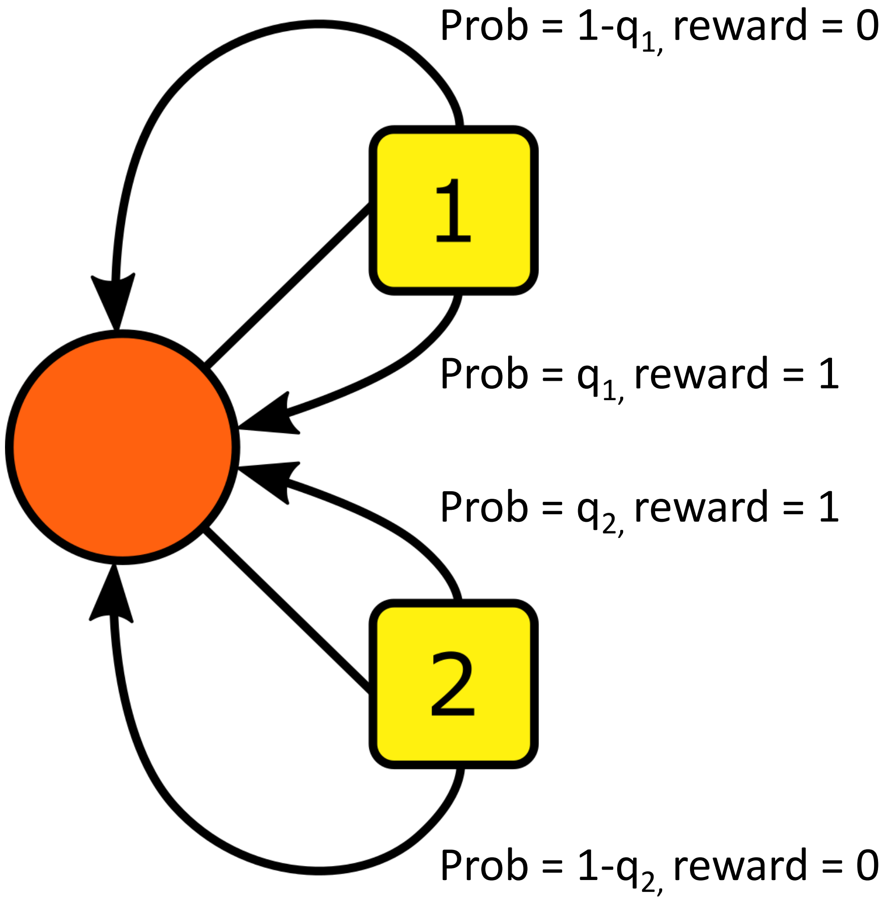
- One state (orange circle).
- Two actions: pull arm 1 or 2.
- Trivial transition probability:

$$p(s|1,s) = p(s|2,s) = 1$$

- Bernoulli reward:

$$r(a=i) = \begin{cases} 1 & w.p \quad q_i \\ 0 & w.p \quad 1-q_i \end{cases}$$

Bandit, hydrogen atom of RL



Utility function:

$$G_\pi = E_\pi\left[\sum_{t=0}^{\infty} \gamma^t r_t\right]$$

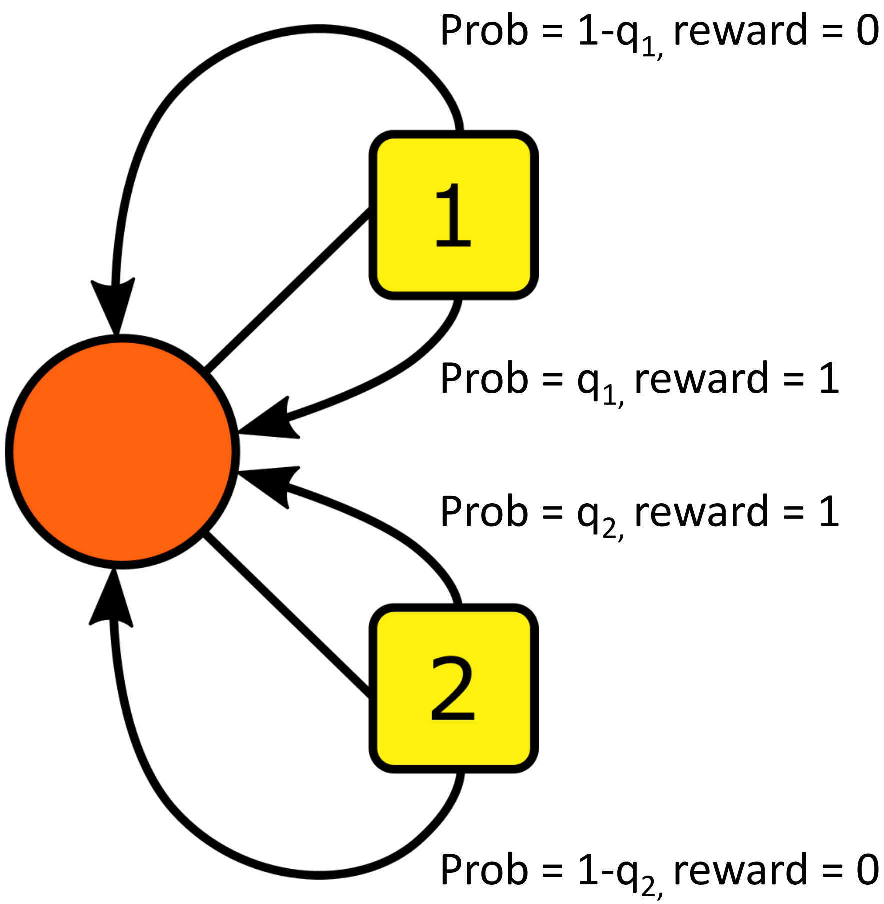
Policy: which action to take

Discount factor

Reward

If q_1 and q_2 are known: always exploit the arm with larger q.

Bandit, hydrogen atom of RL



Utility function:

$$G_\pi = E_\pi\left[\sum_{t=0}^{\infty} \gamma^t r_t\right]$$

Policy: which action to take

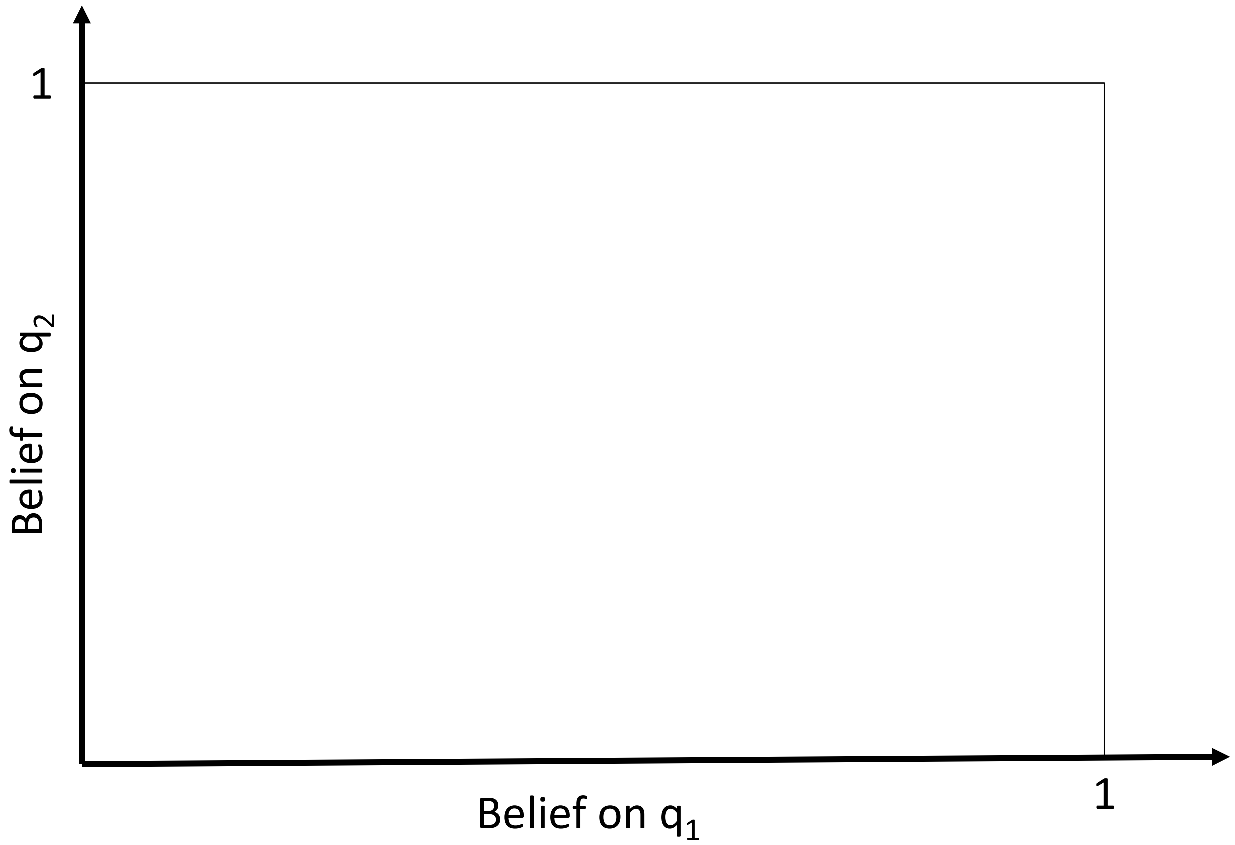
Discount factor

Reward

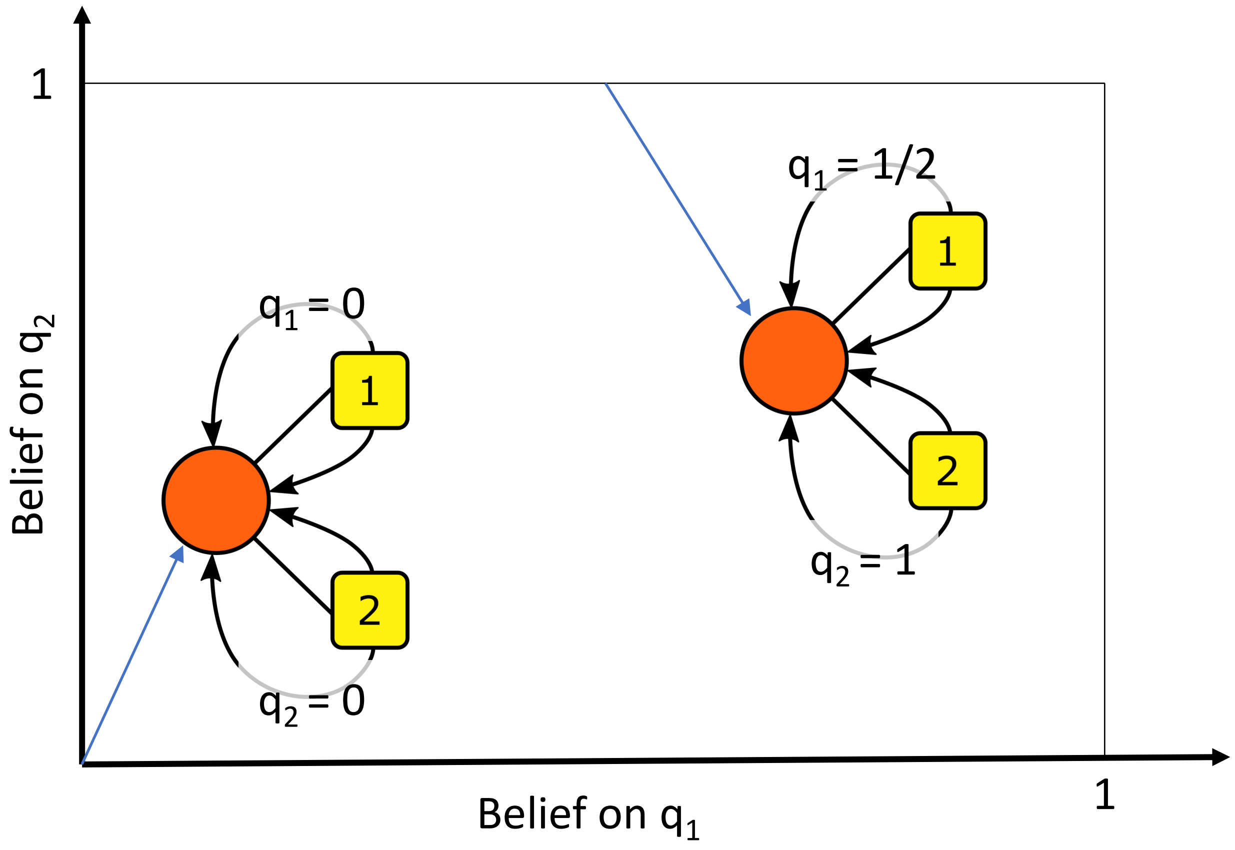
If q_1 and q_2 are unknown???

- **Exploration** -> random choice. *Increase the knowledge of q but I loose chances to get the best reward.*
- **Exploitation** -> pull the best arm. *Without reliable estimates it's useless.*

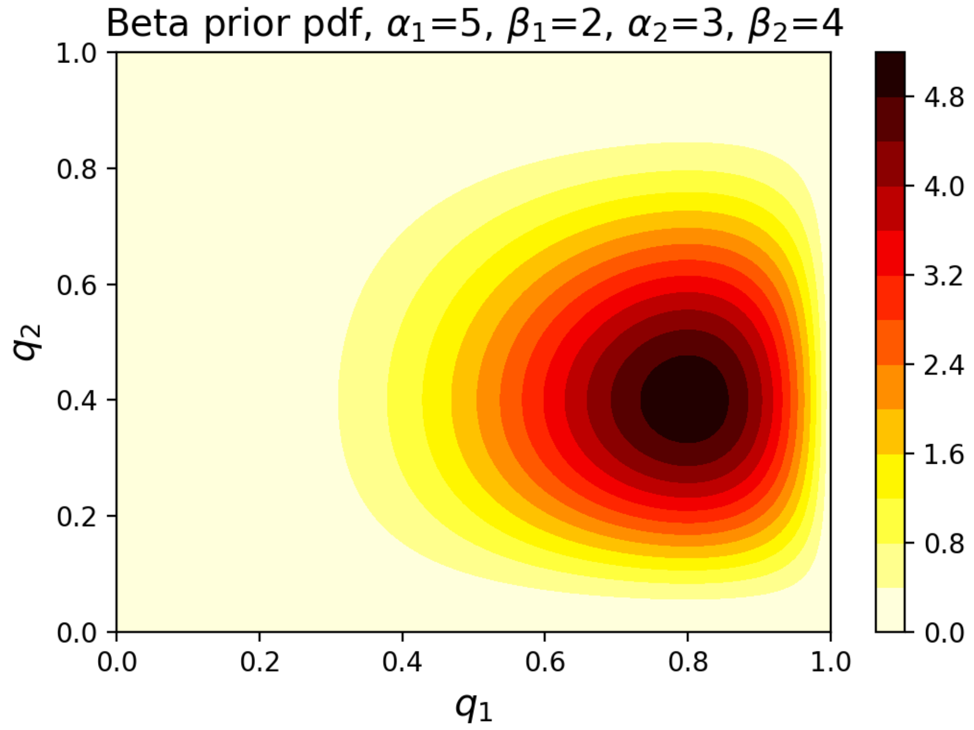
Bandit in the belief space



Bandit in the belief space



Bandit in the belief space



Assumption: the belief is a Beta distribution and the two beliefs are independent:

$$B(q_1, q_2) = Beta(q_1|\alpha_1, \beta_1)Beta(q_2|\alpha_2, b_2\,)$$

Bandit in the belief space

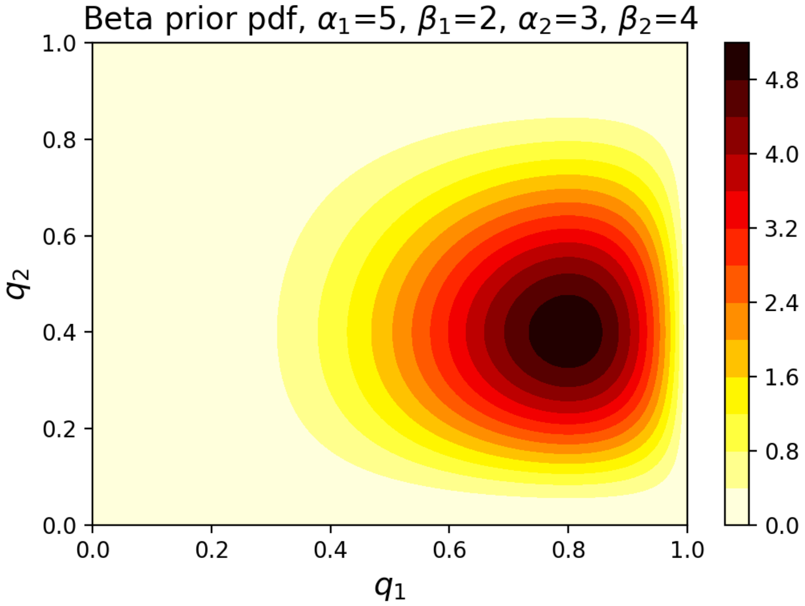
Each time I pull an arm I have a Bernoulli outcome. I can update the belief using Bayes:

$$\underbrace{B_{t+1}(q_1, q_2)}_{\text{Beta again!}} \propto \underbrace{l(r_t|q_1, q_2, a_t)}_{\text{Bernoulli likelyhood}} \underbrace{B_t(q_1, q_2)}_{\text{Beta prior}}$$

Bandit in the belief space

Each time I pull an arm I have a Bernoulli outcome. I can update the beleif using Bayes:

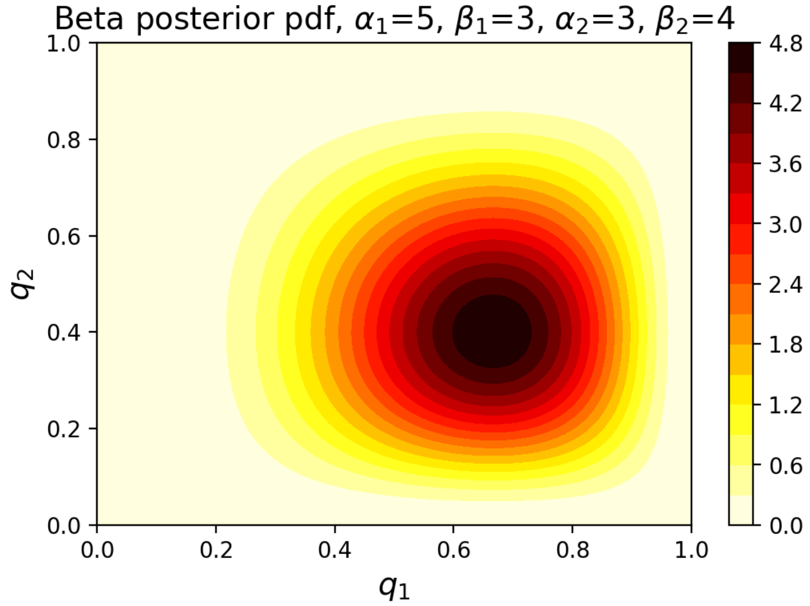
$$B_{t+1}(q_1, q_2) \propto l(r_t | q_1, q_2, a_t) B_t(q_1, q_2)$$



Action: pull arm 1

Not win

Bayes:
$\beta_1 \rightarrow \beta_1 + 1$



Bandit in the belief space

It can be proven that if I start from a flat prior: $Beta_0(q_i|\alpha_i,\beta_i) = Beta_0(q_i|1,1\,) = 1$

I get the following hyperparameters after *t* Bayes updates:

$$\boxed{\alpha_i^{(t)} = n_i^{(t)} + 1} \qquad \boxed{\beta_i^{(t)} = m_i^{(t)} + 1}$$

Number of wins with arm i   Number of losses with arm i

Bandit in the belief space

It can be proven that if I start from a flat prior: $Beta_0(q_i|\alpha_i, \beta_i) = Beta_0(q_i|1,1\,) = 1$

I get the following hyperparameters after *t* Bayes updates:

$$\boxed{\alpha_i^{(t)} = n_i^{(t)} + 1} \qquad \boxed{\beta_i^{(t)} = m_i^{(t)} + 1}$$

Number of wins with arm i   Number of losses with arm i

I build a new MDP game in the belief space

Beleif space $\longrightarrow$ Space of the two Betas $\longrightarrow$ The four hyperparamenters $\alpha_1, \beta_1, \alpha_2, \beta_2$ $\longrightarrow$ The four counters n_1, m_1, n_2, m_2

Belief-space bandit as an MDP

A state is defined by number of wins and losses:

$$s = (n_1, m_1, n_2, m_2)$$

Belief-space bandit as an MDP

A state is defined by number of wins and losses:

$s = (n_1, m_1, n_2, m_2)$

Two actions, the two arms:

$a_t \in \{1,2\}$

Belief-space bandit as an MDP

A state is defined by number of wins and losses:

$$s = (n_1, m_1, n_2, m_2)$$

Two actions, the two arms:

$$a_t \in \{1,2\}$$

Transition probabilities are given by estimates of winning or loosing. If I choose action i:

$$n_i \to n_i + 1 \quad \textit{with prob.} \quad \langle q_i \rangle = \frac{n_i+1}{n_i+m_i+2}$$

$$m_i \to m_i + 1 \quad \textit{with prob.} \quad 1 - \langle q_i \rangle$$

Belief-space bandit as an MDP

A state is defined by number of wins and losses:

$$s = (n_1, m_1, n_2, m_2)$$

Two actions, the two arms:

$$a_t \in \{1,2\}$$

Transition probabilities are given by estimates of winning or loosing. If I choose action i:

$$n_i \to n_i + 1 \quad \textit{with prob.} \quad \langle q_i \rangle = \frac{n_i+1}{n_i+m_i+2}$$

$$m_i \to m_i + 1 \quad \textit{with prob.} \quad 1 - \langle q_i \rangle$$

The reward is the average win, computed with my estimates:

$$r_i = 1\langle q_i \rangle + 0(1 - \langle q_i \rangle) = \frac{n_i + 1}{n_i + m_i + 2}$$

Belief-space bandit as an MDP

A state is defined by number of wins and losses:

$$s = (n_1, m_1, n_2, m_2)$$

Two actions, the two arms:

$$a_t \in \{1,2\}$$

Transition probabilities are given by estimates of winning or loosing. If I choose action i:

$$n_i \to n_i + 1 \qquad \textit{with prob.} \qquad \langle q_i \rangle = \frac{n_i+1}{n_i+m_i+2}$$

$$m_i \to m_i + 1 \qquad \textit{with prob.} \qquad 1 - \langle q_i \rangle$$

The reward is the average win, computed with my estimates:

$$r_i = 1\langle q_i \rangle + 0(1 - \langle q_i \rangle) = \frac{n_i + 1}{n_i + m_i + 2}$$

I have all the ingredients to write down the Bellman equation, solve it:

$$V^*(s) = \max_{a \in \{1,2\}} \sum_{s'} p(s'|s,a)\left[r(s',s) + \gamma V^*(s')\right]$$

Belief-space bandit as an MDP

Two-armed bandit (not knowing the winning probabilities)

↓

Fully observable problem in the belief space.

$S = \{s\}$

↓

$S = \mathbb{Z}^4$

The solution gives me the best action to take given each possible «state of ignorance», i.e. configuration of wins and losses.

Solving with dynamic programming

Each win/loss counter can take infinite values -> **Infinite number of equations**!

Solving with dynamic programming

Each win/loss counter can take infinite values -> **Infinite number of equations**!

- The discount factor introduces a time scale: $P(t > T) = \gamma^T$

Infinite time steps		$(1-\gamma)$ = prob. game stops
$max_\pi \left\{ E_\pi \left[\sum_{t=0}^{\infty} \gamma^t r_t \right] \right\}$	Equivalence	$max_\pi \left\{ E_{\pi,\gamma} \left[\sum_{t=0}^{\infty} r_t \right] \right\}$

Solving with dynamic programming

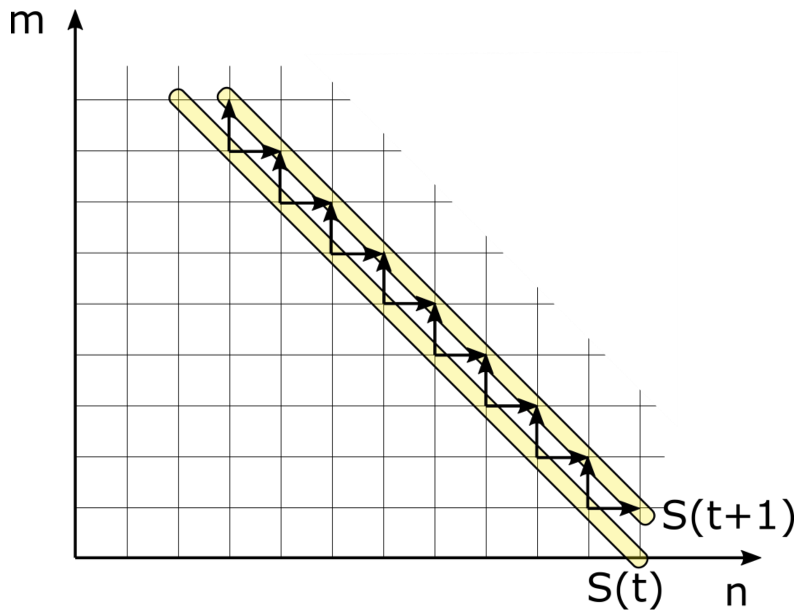
Each win/loss counter can take infinite values -> **Infinite number of equations**!

- The discount factor introduces a time scale: $P(t > T) = \gamma^T$
- At each iteration, the sum of the four counters increases only of one unit, for example:

$$(0,0,0,0) \to (0,1,0,0) \to (0,1,1,0) \to (0,2,1,0) \to \ \dots$$

$$S(t) = \{s = (n_1, m_1, n_2, m_2) \ \text{ such that } \ n_1 + m_1 + n_2 + m_2 = t\}$$

$$S(0) \to S(1) \to S(2) \to \cdots$$



Solving with dynamic programming

Each win/loss counter can take infinite values -> **Infinite number of equations**!

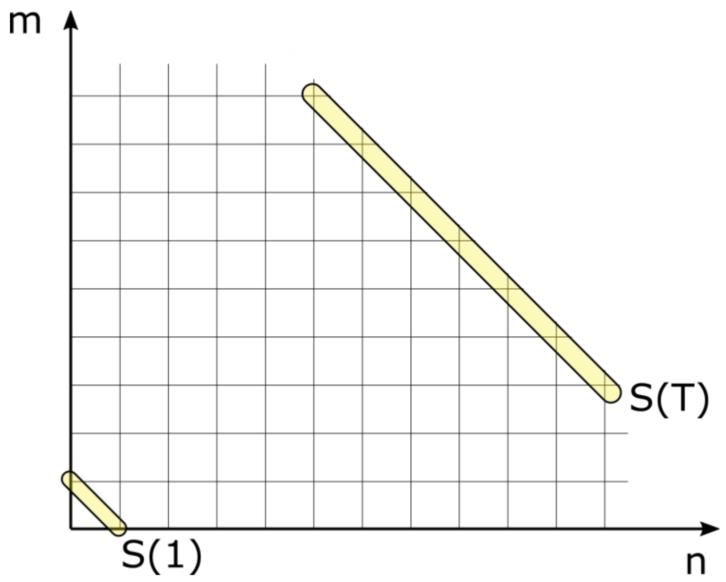
- The discount factor introduce a time scale: $P(t > T) = \gamma^T$
- At each iteration, the sum of the four counters increases only of one unit, for example:

$$(0,0,0,0) \rightarrow (0,1,0,0) \rightarrow (0,1,1,0) \rightarrow (0,2,1,0) \rightarrow \ \ldots$$

I choose *T* such that it is very unlikely that the game lasts for more steps: $P(t > T) = \epsilon, \qquad T = \log \epsilon \ / \ \log \gamma$

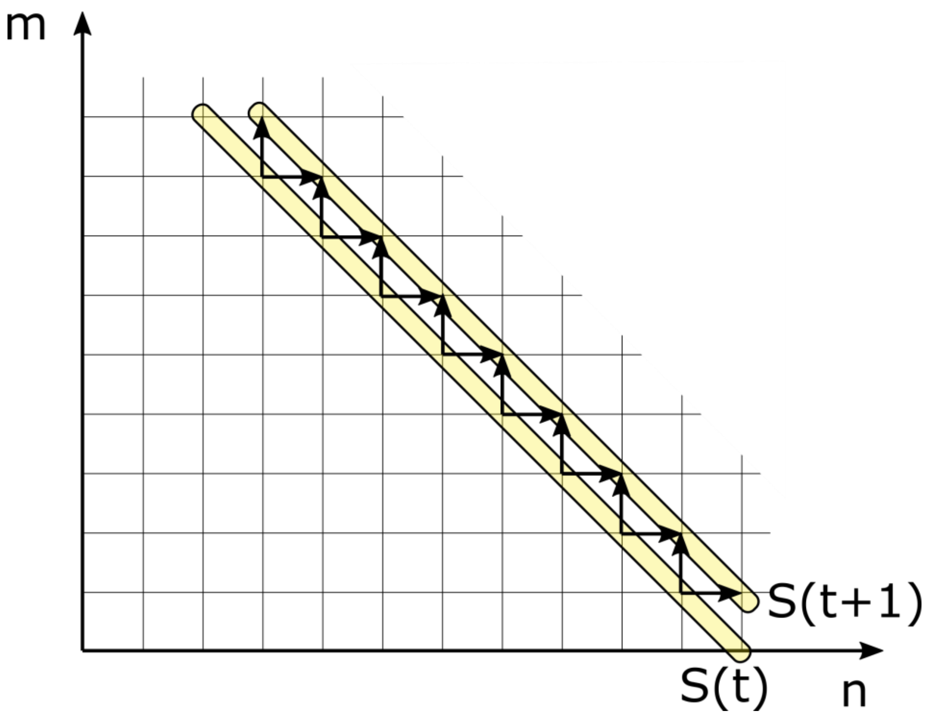
State space approximation:

$$\boxed{\mathcal{S} = \cup_{t<T} S(t)}$$



$$S(t) = \{ s = (n_1, m_1, n_2, m_2) \ \text{ such that } \ n_1 + m_1 + n_2 + m_2 = t \}$$

Solving with dynamic programming



From a state in S(t) I can jump only in S(t+1)

↓

All the values in S(t) can be computed by knowing the values in S(t+1)

$$V^*(s)|_{s\in S(t)} = \max_{a\in\{1,2\}} \sum_{s'\in S(t+1)} p(s'|s,a)\left[r(s',s) + \gamma V^*(s')\right] = B(V^*(s))|_{s\in S(t+1)}$$

Solving with dynamic programming

Solve the Bellman equation going backward:

- Estimate the values at S(T), the boundary, for example the estimated rewards:

$$V\big(s \in S(T)\big) = max\{\langle q_1(s)\rangle, \langle q_2(s)\rangle\}$$

- Iteratively compute all the values going backward.

Solving with dynamic programming

Solve the Bellman equation going backward:

- Estimate the values at S(T), the boundary, for example the estimated rewards:

$$V\big(s \in S(T)\big) = max\{\langle q_1(s)\rangle, \langle q_2(s)\rangle\}$$

- Iteratively compute all the values going backward.

Even though the estimate at the boundary is wrong, the error exponentially disappears:

Error of the states in S(t) $\sim \gamma^{T-t}$

Solving with dynamic programming

$$V\big(s \in S(t)\big) = max_a E[r + \boldsymbol{\gamma} V(s\epsilon S(t+1)] = max_a\{E[r] + E[\boldsymbol{\gamma} V(s\epsilon S(t+1)]\} =$$

$$= something \quad + \quad \boldsymbol{\gamma} \cdot a\ function\ of\ V(s\epsilon S(t+1))$$

$$= something \quad + \quad \boldsymbol{\gamma} \cdot something\ +\ \dots\ + \quad \boldsymbol{\gamma}^{T-t} a\ function\ of\ V(s\epsilon S(T))$$

Even though the estimate at the boundary is wrong, the error exponentially disappears:

Error of the states in S(t) $\sim\ \gamma^{T-t}$