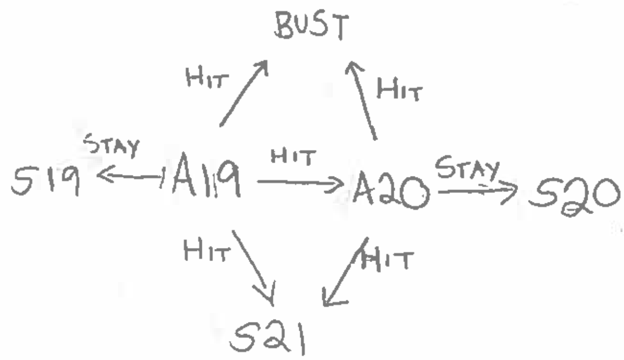
Markov Decision Process

1) Rather than treating blackjack as a 2-player game where one player is Fate, we could model it as the following (nondeterministic) state machine:



i.e. $M = (Q, \Sigma, \Delta, q_0, F)$ where:

- $Q = \{A19, S19, A20, S20, S21, BUST\}$
- $\Sigma = \{HIT, STAY\}$
- $\Delta = \{(A19, STAY, S19), (A20, STAY, S20), (A19, HIT, A20), (A19, HIT, S21), (A19, HIT, BUST), (A20, HIT, S21), (A20, HIT, BUST)\}$
- $q_0 = A19$
- $F = \{S19, S20, S21, BUST\}$

---

2) In addition, we want to specify a reward R_t for reaching a state after t transitions:

$\forall t \geq 0$

$R_t(S20) = .58$	$R_t(A20) = 0$
$R_t(S21) = .88$	$R_t(BUST) = -1$
$R_t(S19) = .27$	$R_t(A19) = 0$

3) And finally, we want to know the likelihood of each transition, given we take a particular action in a particular state.

$$w(\langle 19, \text{STAY}, S19\rangle) = 1$$
$$w(\langle 20, \text{STAY}, S20\rangle) = 1$$

$$w(\langle 19, \text{HIT}, q'\rangle) = P(q' \mid q = 19, \sigma = \text{HIT})$$

$$w(\langle 19, \text{HIT}, 20\rangle) = \frac{1}{13}$$
$$w(\langle 19, \text{HIT}, S21\rangle) = \frac{1}{13}$$
$$w(\langle 19, \text{HIT}, \text{BUST}\rangle) = \frac{11}{13}$$

$$\sum_{q'} w(\langle 19, \text{HIT}, q'\rangle) = 1$$

$$w(\langle 20, \text{HIT}, S21\rangle) = \frac{1}{13}$$
$$w(\langle 20, \text{HIT}, \text{BUST}\rangle) = \frac{12}{13}$$

$$\sum_{q'} w(\langle 20, \text{HIT}, q'\rangle) = 1$$

4) This is called a Markov Decision Process (MDP). Formally an MDP is a triple (M, R, w), where:

- $M = (Q, \Sigma, \Delta, q_0, F)$ is a state machine
- $R : Q \times \mathbb{N} \to \mathbb{R}$ is a "reward function"
- $w : \Delta \mapsto \mathbb{R}$ s.t. for all $q \in Q$, $\sigma \in \Sigma$: $\sum_{(q, \sigma, q') \in \Delta} w(\langle q, \sigma, q'\rangle) = 1$

We assume that $R(q, t) = \gamma \cdot R(q, t-1)$ $\forall q \in Q$, $t \geq 1$ for some "discounting factor" γ s.t. $0 < \gamma \leq 1$.

MARKOV DECISION PROCESS

5) It can be helpful to define the following shorthand for dealing with MDPs:

→ $R_t(q) \triangleq R(q, t) \qquad \forall q \in Q,\ t \geq 0$

→ $P(q' \mid q, \sigma) \triangleq \omega(\langle q, \sigma, q' \rangle) \qquad \forall \langle q, \sigma, q' \rangle \in \Delta$

→ We write path $\langle (q, \sigma_0, q_1), (q_1, \sigma_1, q_2), \ldots, (q_{n-1}, \sigma_{n-1}, q_n) \rangle$ as:

$$q \xrightarrow{\sigma_0} q_1 \xrightarrow{\sigma_1} \cdots \xrightarrow{\sigma_{n-1}} q_n$$

→ $\text{reward}\left(q \xrightarrow{\sigma_0} q_1 \xrightarrow{\sigma_1} \cdots \xrightarrow{\sigma_{n-1}} q_n\right)$

$$= R_0(q) + R_1(q_1) + \ldots + R_n(q_n)$$

→ $P\left(q \xrightarrow{\sigma_0} q_1 \xrightarrow{\sigma_1} \cdots \xrightarrow{\sigma_{n-1}} q_n\right)$

$$= P(q_1 \mid q, \sigma_0) \cdot P(q_2 \mid q_1, \sigma_1) \cdot \ldots \cdot P(q_n \mid q_{n-1}, \sigma_{n-1})$$

---

6) The main computational challenge, given an MDP, is to determine the best decision to make in each state.

For example:

If I have 19, should I HIT or STAY?
If I have 20, should I HIT or STAY?

We can formalize this as a function $\pi : (Q \setminus F) \to \Sigma$, which we call a <u>policy</u>.

e.g. $\pi(A19) = \text{HIT}$
$\pi(A20) = \text{HIT}$

Markov Decision Process

7) Given a policy π, I can compute my expected reward U^π, starting from various states:

$$U^\pi(A19) = \text{reward}(A19 \xrightarrow{HIT} A20 \xrightarrow{HIT} S21)\,P(A19 \xrightarrow{HIT} A20 \xrightarrow{HIT} S21)$$
$$+ \text{reward}(A19 \xrightarrow{HIT} A20 \xrightarrow{HIT} BUST)\,P(A19 \xrightarrow{HIT} A20 \xrightarrow{HIT} BUST)$$
$$+ \text{reward}(A19 \xrightarrow{HIT} S21)\,P(A19 \xrightarrow{HIT} S21)$$
$$+ \text{reward}(A19 \xrightarrow{HIT} BUST)\,P(A19 \xrightarrow{HIT} BUST)$$

$$U^\pi(A20) = \text{reward}(A20 \xrightarrow{HIT} S21)\,P(A20 \xrightarrow{HIT} S21)$$
$$+ \text{reward}(A20 \xrightarrow{HIT} BUST)\,P(A20 \xrightarrow{HIT} BUST)$$

---

8) These expected rewards (usually called <u>expected utility</u>) can be expressed in terms of each other:

$$U^\pi(A19) = \left(R_0(A19) + R_1(A20) + R_2(S21)\right)P(A19 \xrightarrow{HIT} A20 \xrightarrow{HIT} S21)$$
$$+ \left(R_0(A19) + R_1(A20) + R_2(BUST)\right)P(A19 \xrightarrow{HIT} A20 \xrightarrow{HIT} BUST)$$
$$+ \left(R_0(A19) + R_1(S21)\right)P(A19 \xrightarrow{HIT} S21)$$
$$+ \left(R_0(A19) + R_1(BUST)\right)P(A19 \xrightarrow{HIT} BUST)$$

$$= R_0(A19)\Big[P(A19 \xrightarrow{HIT} A20 \xrightarrow{HIT} S21) + P(A19 \xrightarrow{HIT} A20 \xrightarrow{HIT} BUST)$$
$$+ P(A19 \xrightarrow{HIT} S21) + P(A19 \xrightarrow{HIT} BUST)\Big]$$
$$+ \left(R_1(A20) + R_2(S21)\right)P(A19 \xrightarrow{H} A20 \xrightarrow{H} S21)$$
$$+ \left(R_1(A20) + R_2(BUST)\right)P(A19 \xrightarrow{H} A20 \xrightarrow{H} BUST)$$
$$+ R_1(S21)\,P(A19 \xrightarrow{H} S21)$$
$$+ R_1(BUST)\,P(A19 \xrightarrow{H} BUST)$$

this equals 1, b/c it is a prob. distribution.

Markov Decision Process

(8) (cont.)

$$
\begin{aligned}
U^{\pi}(A19) &= R_0(A19) \\
&\quad + P(A20 \mid A19, \text{HIT}) \Big[\big(R_1(A20) + R_2(S21)\big) P(A20 \xrightarrow{H} S21) \\
&\qquad + \big(R_1(A20) + R_2(\text{BUST})\big) P(A20 \xrightarrow{H} \text{BUST}) \Big] \\
&\quad + P(S21 \mid A19, \text{HIT})\, R_1(S21) \\
&\quad + P(\text{BUST} \mid A19, \text{HIT})\, R_1(\text{BUST}) \\
&= R_0(A19) \\
&\quad + P(A20 \mid A19, \text{HIT}) \Big[\big(\gamma R_0(A20) + \gamma R_1(S21)\big) P(A20 \xrightarrow{H} S21) \\
&\qquad + \big(\gamma R_0(A20) + \gamma R_1(\text{BUST})\big) P(A20 \xrightarrow{H} \text{BUST}) \Big] \\
&\quad + P(S21 \mid A19, \text{HIT}) \big(\gamma \cdot R_0(S21)\big) \\
&\quad + P(\text{BUST} \mid A19, \text{HIT}) \big(\gamma \cdot R_0(\text{BUST})\big) \\
&= R_0(A19) \\
&\quad + \gamma \cdot P(A20 \mid A19, \text{HIT}) \Big[\overbrace{\big(R_0(A20) + R_1(S21)\big)}^{\text{reward}(A20 \xrightarrow{H} S21)} P(A20 \xrightarrow{H} S21) \\
&\qquad + \underbrace{\big(R_0(A20) + R_1(\text{BUST})\big)}_{\text{reward}(A20 \xrightarrow{H} \text{BUST})} P(A20 \xrightarrow{H} \text{BUST}) \Big] \\
&\quad + \gamma \cdot P(S21 \mid A19, \text{HIT})\, R_0(S21) \\
&\quad + \gamma \cdot P(\text{BUST} \mid A19, \text{HIT})\, R_0(\text{BUST}) \\
&= R_0(A19) \\
&\quad + \gamma \cdot P(A20 \mid A19, \text{HIT})\, U^{\pi}(A20) \\
&\quad + \gamma \cdot P(S21 \mid A19, \text{HIT})\, U^{\pi}(S21) \\
&\quad + \gamma \cdot P(\text{BUST} \mid A19, \text{HIT})\, U^{\pi}(\text{BUST}) \\
&= R_0(A19) \\
&\quad + \gamma \cdot \Big[U^{\pi}(A20) \cdot P(A20 \mid A19, \pi(A19)) \\
&\qquad + U^{\pi}(S21) \cdot P(S21 \mid A19, \pi(A19)) \\
&\qquad + U^{\pi}(\text{BUST}) \cdot P(\text{BUST} \mid A19, \pi(A19)) \Big] \\
&= R_0(A19) + \gamma \cdot \sum_{q' \in Q} U^{\pi}(q') \cdot P(q' \mid A19, \pi(A19))
\end{aligned}
$$

Markov Decision Process

⑨ This is a general result:

$$U^{\pi}(q) = R_0(q) + \gamma \sum_{q' \in Q} U^{\pi}(q') \cdot P(q' \mid q, \pi(q))$$

⑩ Usually we're not simply interested in computing the expected utility of a state, given some arbitrary policy π. Rather, we'd like to know how much reward we should expect if we execute the <u>best</u> policy π^*.

$$\begin{aligned}
U(q) &= \max_{\pi} U^{\pi}(q) \\
&= \max_{\pi} \left[R_0(q) + \gamma \sum_{q' \in Q} U^{\pi}(q') \cdot P(q' \mid q, \pi(q)) \right] \\
&= R_0(q) + \gamma \cdot \max_{\pi} \sum_{q' \in Q} U^{\pi}(q') \cdot P(q' \mid q, \pi(q)) \\
&= R_0(q) + \gamma \cdot \max_{\sigma} \max_{\pi \mid \pi(q) = \sigma} \sum_{q' \in Q} U^{\pi}(q') \cdot P(q' \mid q, \sigma)
\end{aligned}$$

we choose some action σ for state q

$$\begin{aligned}
&= R_0(q) + \gamma \cdot \max_{\sigma} \sum_{q' \in Q} \left(\max_{\pi \mid \pi(q) = \sigma} U^{\pi}(q') \right) \cdot P(q' \mid q, \sigma) \\
&= \underbrace{R_0(q)}_{\text{reward from the current state}} + \underbrace{\gamma \cdot \max_{\sigma} \sum_{q' \in Q} U(q') \cdot P(q' \mid q, \sigma)}_{\text{maximum expected utility of the next state, given optimal action}}
\end{aligned}$$

Markov Decision Process

(11) For our blackjack example, we get the following equations.

$$U(A19) = \max \left\{ \begin{array}{l} U(S19) \cdot P(S19 \mid A19, \text{STAY}), \\ U(A20) \cdot P(A20 \mid A19, \text{HIT}) \\ \quad + U(S21) \cdot P(S21 \mid A19, \text{HIT}) \\ \quad + U(\text{BUST}) \cdot P(\text{BUST} \mid A19, \text{HIT}) \end{array} \right\}$$

$$U(A20) = \max \left\{ \begin{array}{l} U(S20) \cdot P(S20 \mid A20, \text{STAY}), \\ U(S21) \cdot P(S21 \mid A20, \text{HIT}) \\ \quad + U(\text{BUST}) \cdot P(\text{BUST} \mid A20, \text{HIT}) \end{array} \right\}$$

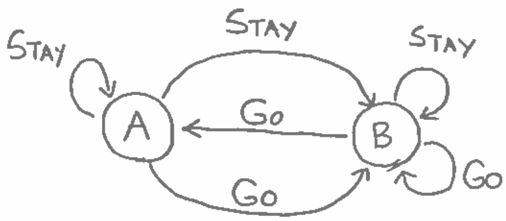
$$U(S19) = R_o(S19)$$
$$U(S20) = R_o(S20)$$
$$U(S21) = R_o(S21)$$
$$U(\text{BUST}) = R_o(\text{BUST})$$

We have six equations for six unknowns ($U(A19)$, $U(A20)$, $U(S19)$, $U(S20)$, $U(S21)$, $U(\text{BUST})$).

(12) Importantly, these equations are not linear, so we can't use linear algebra techniques.

MARKOV DECISION PROCESS

(13) To motivate our equation-solving technique, let's use a smaller MDP:



where:

$$P(A \mid A, \text{STAY}) = \frac{1}{2} \qquad P(B \mid B, \text{STAY}) = 1$$

$$P(B \mid A, \text{STAY}) = \frac{1}{2} \qquad P(A \mid B, \text{Go}) = \frac{1}{5}$$

$$P(B \mid A, \text{Go}) = 1 \qquad P(B \mid B, \text{Go}) = \frac{4}{5}$$

and

$$R_t(A) = \gamma^t \cdot 1$$

$$R_t(B) = \gamma^t \cdot (-1)$$

where $\gamma = \frac{1}{2}$.

(14) We get the following equations:

$$U(A) = R_0(A) + \gamma \cdot \max \left\{ U(B),\ U(A)P(A \mid A, \text{STAY}) + U(B)P(B \mid A, \text{STAY}) \right\}$$

$$= 1 + \frac{1}{2} \max \left\{ U(B),\ \frac{1}{2}U(A) + \frac{1}{2}U(B) \right\}$$

$$U(B) = R_0(B) + \gamma \cdot \max \left\{ U(B),\ U(A)P(A \mid B, \text{Go}) + U(B)P(B \mid B, \text{Go}) \right\}$$

$$= -1 + \frac{1}{2} \max \left\{ U(B),\ \frac{1}{5}U(A) + \frac{4}{5}U(B) \right\}$$

MARKOV DECISION PROCESS

(15) Suppose we guess the values of $U(A)$ and $U(B)$:

U_0^A is our guess at $U(A)$

U_0^B is our guess at $U(B)$

Consider the following iterative algorithm:

for $i = 1$ to ∞:

let $U_i^A = 1 + \frac{1}{2} \max\left\{U_{i-1}^B, \frac{1}{2}U_{i-1}^A + \frac{1}{2}U_{i-1}^B\right\}$

let $U_i^B = -1 + \frac{1}{2} \max\left\{U_{i-1}^B, \frac{1}{5}U_{i-1}^A + \frac{4}{5}U_{i-1}^B\right\}$

At each iteration, we assume our guesses for $U(A)$ and $U(B)$ from the previous iteration are correct, and we compute new guesses using our equations.

---

(16) Why would this ever work? Well, it seems to converge...

i	U_i^A	U_i^B
0	0	0
1	1	-1
2	1	-1.3
3	.925	-1.42
4	.876	-1.476
5	.850	-1.503
6	.836	-1.516
7	.830	-1.523

(17) But can we prove it? Assume for the moment that a solution exists, i.e. there's a vector

$$U^* = \begin{bmatrix} U^A \\ U^B \end{bmatrix}$$

such that:

$$U^A = 1 + \frac{1}{2} \max \left\{ U^B, \frac{1}{2} U^A + \frac{1}{2} U^B \right\}$$
$$U^B = -1 + \frac{1}{2} \max \left\{ U^B, \frac{1}{5} U^A + \frac{4}{5} U^B \right\}$$

(18) Let's also measure how bad our guesses are. Suppose the solution is $\begin{bmatrix} 2 \\ 4 \end{bmatrix}$ and our guess is $\begin{bmatrix} 5 \\ 3 \end{bmatrix}$.

We'll measure the distance of our guesses to the solution as the absolute difference between our worst guess and the solution, i.e.

$$\text{dist}\left(\begin{bmatrix} 5 \\ 3 \end{bmatrix}, \begin{bmatrix} 2 \\ 4 \end{bmatrix} \right) = \max \left\{ |5-2|, |3-4| \right\} = 3$$

(19) Now what if we could show that our guesses get better every iteration? i.e. that

$$\text{dist}\left(\begin{bmatrix} U_{i+1}^A \\ U_{i+1}^B \end{bmatrix}, \begin{bmatrix} U^A \\ U^B \end{bmatrix}\right) \leq K \cdot \text{dist}\left(\begin{bmatrix} U_i^A \\ U_i^B \end{bmatrix}, \begin{bmatrix} U^A \\ U^B \end{bmatrix}\right)$$

for $0 \leq K < 1$.

That would mean:

$$\lim_{i \to \infty} \text{dist}\left(\begin{bmatrix} U_i^A \\ U_i^B \end{bmatrix}, \begin{bmatrix} U^A \\ U^B \end{bmatrix}\right)$$

$$\leq \lim_{i \to \infty} K^i \, \text{dist}\left(\begin{bmatrix} U_0^A \\ U_0^B \end{bmatrix}, \begin{bmatrix} U^A \\ U^B \end{bmatrix}\right)$$

$$\equiv \lim_{i \to \infty} K^i$$

$$= 0$$

Since $\text{dist}\left(\begin{bmatrix} U_i^A \\ U_i^B \end{bmatrix}, \begin{bmatrix} U^A \\ U^B \end{bmatrix}\right) \geq 0$, thus $\lim_{i \to \infty} \text{dist}\left(\begin{bmatrix} U_i^A \\ U_i^B \end{bmatrix}, \begin{bmatrix} U^A \\ U^B \end{bmatrix}\right) = 0$

So our guesses would converge to the solution.

MARKOV DECISION PROCESS

(20) So then let's show it.

$$\text{dist}\left(\begin{bmatrix} U_{i+1}^A \\ U_{i+1}^B \end{bmatrix}, \begin{bmatrix} U^A \\ U^B \end{bmatrix}\right) = \max_{q \in \{A,B\}} \left| U_{i+1}^q - U^q \right|$$

If we simplify $|U_{i+1}^q - U^q|$, we get:

$$\left| U_{i+1}^q - U^q \right| = \left| \gamma \cdot \left[\max_\sigma \sum_{q'} U_i^{q'} P(q' | q, \sigma) - \max_\sigma \sum_{q'} U^{q'} P(q' | q, \sigma) \right] \right|$$

because $\forall f, g$ $\left|\max_\sigma f(\sigma) - \max_\sigma g(\sigma)\right| \leq \max_\sigma |f(\sigma) - g(\sigma)|$

$$\leq \gamma \cdot \max_\sigma \left| \sum_{q'} U_i^{q'} P(q' | q, \sigma) - \sum_{q'} U^{q'} P(q' | q, \sigma) \right|$$

$$= \gamma \cdot \max_\sigma \left| \sum_{q'} \left(U_i^{q'} - U^{q'} \right) P(q' | q, \sigma) \right|$$

$$\leq \gamma \cdot \max_\sigma \sum_{q'} \left| U_i^{q'} - U^{q'} \right| P(q' | q, \sigma)$$

$$\leq \gamma \cdot \max_\sigma \max_{q'} \left| U_i^{q'} - U^{q'} \right|$$

(because the weighted average of a set of numbers is at most the max)

$$= \gamma \cdot \max_{q'} \left| U_i^{q'} - U^{q'} \right|$$

Thus:

$$\text{dist}\left(\begin{bmatrix} U_{i+1}^A \\ U_{i+1}^B \end{bmatrix}, \begin{bmatrix} U^A \\ U^B \end{bmatrix}\right) \leq \max_q \gamma \cdot \max_{q'} \left| U_i^{q'} - U^{q'} \right|$$

$$= \gamma \max_{q'} \left| U_i^{q'} - U^{q'} \right|$$

$$= \gamma \cdot \text{dist}\left(\begin{bmatrix} U_i^A \\ U_i^B \end{bmatrix}, \begin{bmatrix} U^A \\ U^B \end{bmatrix}\right)$$

MARKOV DECISION PROCESS

(21) So as long as the discounting factor is between 0 and 1 (not including 1), then our iterative technique (called "value iteration") will converge to the correct solution.