Acing Blackjack
- A Monte Carlo approach

Blackjack 101

- Blackjack (also known as 21) is the most widely played casino game in the world.
- It is a comparing card game played between player(s) and the dealer.
    - Players compete against the dealer but not against each other.
- The objective of the game is to outscore the dealer (via sum of dealt cards) without exceeding 21.

Blackjack gameplay

- The player is dealt a two-card hand
  - **Face cards** (Kings, Queens, and Jacks) are counted as **10** points.
  - **Aces** may be counted as **1 or 11** points.
  - All other cards are counted as the numeric value shown on the card.
- The dealer then deals two cards from the deck for himself and shows only the first card (face-up) to the player. The second card is kept face-down.
- After looking at the dealer’s face-up card, the player may decide to "**hit**“, i.e., receiving additional card(s) one-by-one until
    - The card sum crosses 21 (player is “**busted**”).
    - Player decides to end his turn (player “**sticks**”).
- Once the player sticks, it becomes the dealer’s turn.
  - The **dealer** plays with a **fixed strategy** without choice: he continues to hit until his cards total **17** or more points.
- Finally, the player’s and dealer’s card sums are computed.

Blackjack: ways to win

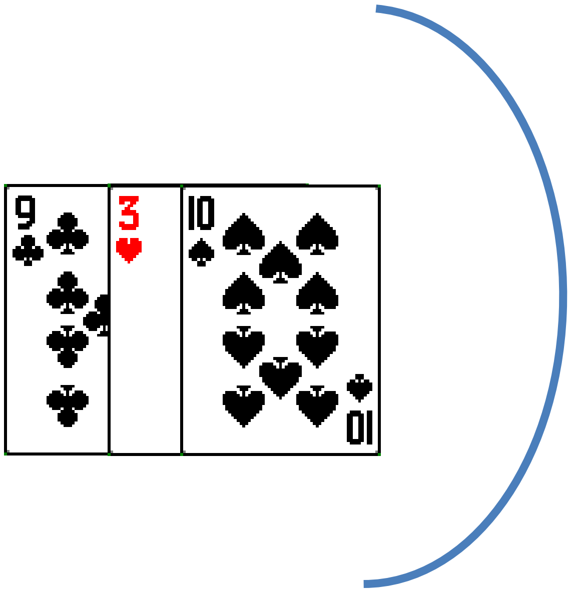
- Three ways to beat the dealer:
  - Get 21 points on the player's first two cards itself (called a **natural**), without a dealer natural. The game ends immediately in this case.
  - Reach a final score higher than the dealer without exceeding 21 (otherwise **player busted**).
  - Let the dealer draw additional cards until his hand exceeds 21 (**dealer busted**).
- If the dealer and player hand sums are the same, it’s a tie!

Blackjack 201

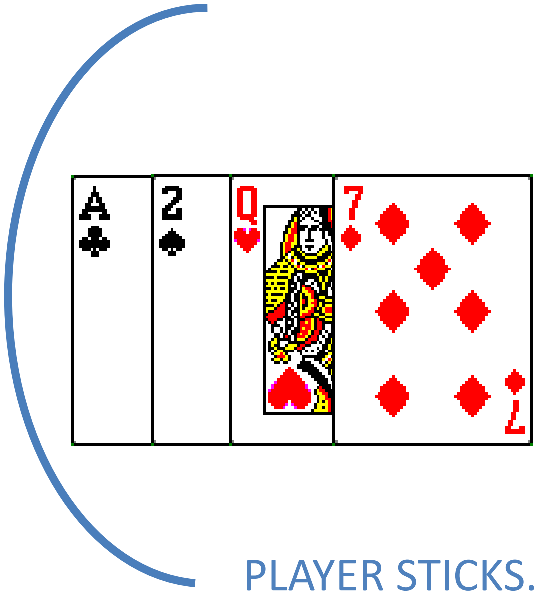
- If the player holds an ace that he could count as 11 without going bust, then the ace is said to be **usable** (also called “**Soft Hand**”). Otherwise, its called **hard**. The term soft means the hand has two possible totals.
  - In this case it is always counted as 11 because counting it as 1 would make the sum 11 or less, in which case there is no decision to be made because, obviously, the player should always hit.
- We only consider “Hit” and “Stick” here. Other options include
  - “**Double Down**” (double wager, take a single card and finish).
  - “**Split**” (if the two cards have the same value, separate them to make two hands).
  - “**Surrender**”(give up a half-bet and retire from the game).
- Most blackjack tables have a **payout** of **3:2**.

Blackjack sample gameplay

DEALER



PLAYER



PLAYER STICKS.

DEALER SUM 12; NO USABLE ACE
DEALER SUM 22; NO USABLE ACE

PLAYER SUM: 13; USABLE ACE
PLAYER SUM: 20; NO USABLE ACE

Dealer busted => Player wins!

The Problem

- What is the optimal winning strategy (policy)?

  i.e., When do you hit? When do you stick?
- **Markov decision process** provide a mathematical framework for this environment:
  - The player and dealer cards constitute the current **state.**
  - The possible **actions** from any state are to hit or stick.
  - Transition **probabilities** represent the probabilities of moving to the next state.
  - At the end of game, the player gets a **reward** (win/loss).
- Finding the optimal state value or action value function gives the solution to this problem.

Dynamic Programing (DP)

- DP requires complete model of the environment as a Markov decision process (states and possible actions/transition probabilities/expected immediate rewards).
- Optimal state-value function is

$$\begin{aligned} V^*(s) &= \max_a E\left\{r_{t+1} + \gamma V^*(s_{t+1}) \mid s_t = s, a_t = a\right\} \\ &= \max_a \sum_{s'} \boxed{\mathcal{P}^a_{ss'}} \left[\boxed{\mathcal{R}^a_{ss'}} + \gamma V^*(s')\right] \end{aligned}$$

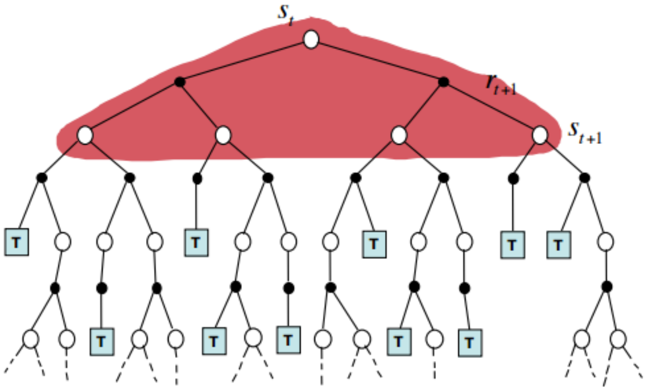
- DP methods require the distribution of next events which is not easy to determine in the case of blackjack.
  - For example, suppose the player's sum is 18 and he chooses to stick. What is his expected reward and transition probabilities? These computations are often complex and error-prone.

Monte Carlo (MC)

- Unlike DP methods, MC methods do not assume complete knowledge of the environment.
- MC methods require only *experience*--sample sequences of states, actions, and rewards from on-line or simulated interaction with an environment.
  - By trading-off exploration (in unknown territory) and exploitation (in known territory), MC methods can achieve optimal performance.

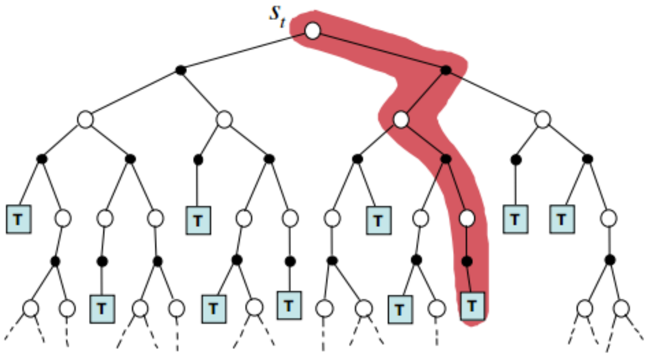
DP versus MC

$$V(s_t) \leftarrow E_\pi\{r_{t+1} + \gamma V(s_{t+1})\}$$



$$V(s_t) \leftarrow V(s_t) + \alpha\left[R_t - V(s_t)\right]$$

where R_t is the actual return following state s_t.



Value of a state depends on the average reward, the transition probabilities and the value of other states

Value of a state depends only on the final return

Advantages of MC over DP

- MC methods can be used to learn optimal behavior directly from interaction with the environment, with **no model of the environment's dynamics**.
- Monte Carlo methods are particularly attractive when one requires the **values of only a subset of states**.
  - One can generate many sample episodes starting from these states alone, averaging returns only from these states ignoring all others.
- MC methods **do not “bootstrap”**: The estimate for one state does not build upon the estimate of any other state, as is the case in DP (or temporal difference learning methods).
- **Not very sensitive** to initial values; easy to understand and use.
- MC methods work in **non-Markovian** environments as well.

Solving Blackjack

- Playing Blackjack is naturally formulated as an *episodic* finite MDP. Each game of blackjack is an "episode".
  - We assume experience is divided into episodes, and that all episodes eventually terminate no matter what actions are selected.

STATES & ACTIONS

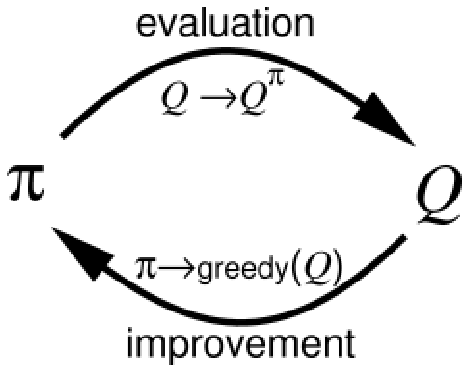
- The states depend on the player's cards and the dealer's showing card.
- Basically, the player makes decisions on the basis of three variables: his current sum (12-21), the dealer's one showing card (ace-10), and whether or not he holds a usable ace. This makes for a total of (10*10*2= 200) states.
- Possible actions from any state are **hit** and **stick**.

RETURNS

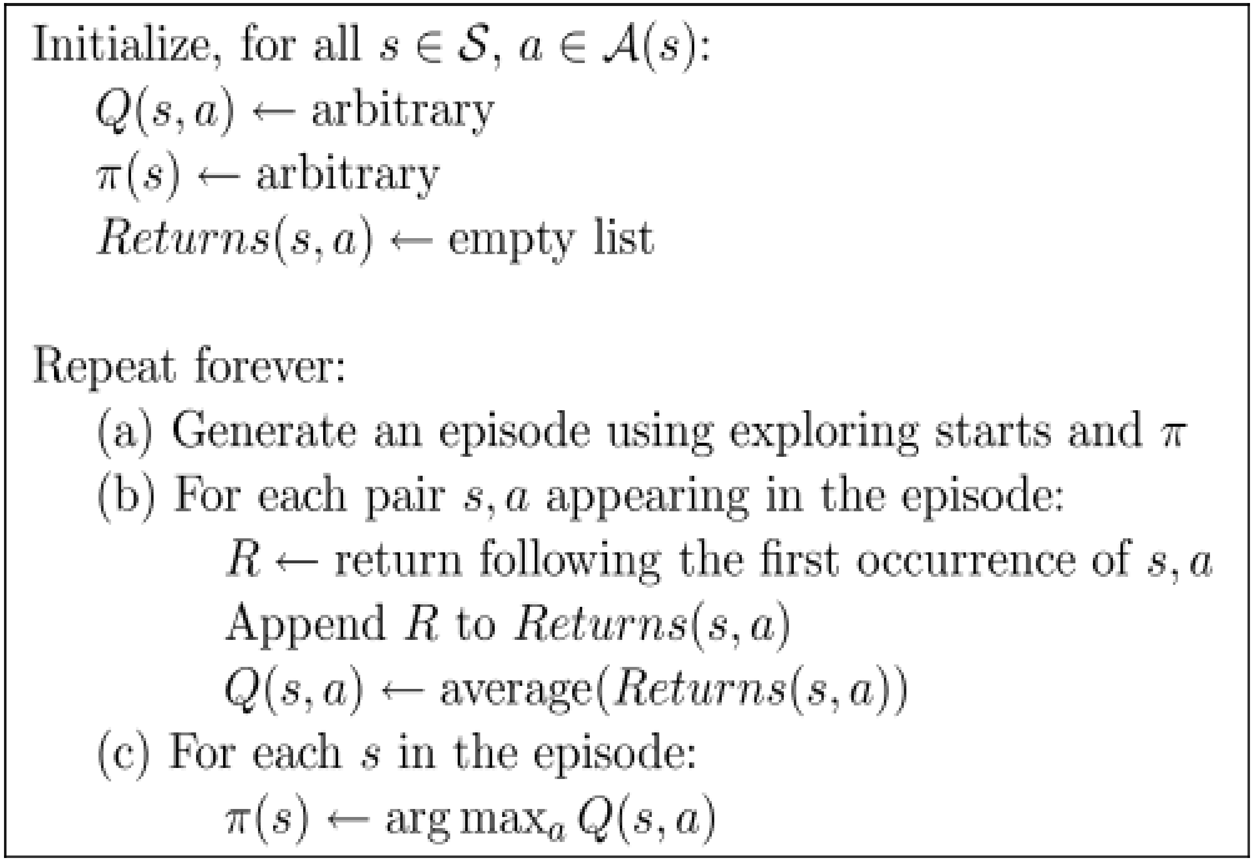
- Returns of 1.5$, -1$, and 0$ are given for winning, losing, and drawing, respectively only at the end of the episode for every $ bet. Note: they are not the immediate expected rewards.
- All rewards within a game are zero, and we do not discount ($\gamma = 1$); therefore these terminal rewards are also the returns.

Monte Carlo control

- Finding the optimal policy (strategy) classically comprises a generalized form of **policy iteration.**
  - Start off with an arbitrary strategy π and **evaluate** its state/action values (Q).
  - **Improve** the strategy in a 'greedy' manner.
  - **Repeat** the policy evaluation and improvement steps until the policy and action value functions attain optimality.



Monte Carlo with Exploring Starts



```
Initialize, for all $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$:
    $Q(s,a) \leftarrow$ arbitrary
    $\pi(s) \leftarrow$ arbitrary
    $Returns(s,a) \leftarrow$ empty list

Repeat forever:
    (a) Generate an episode using exploring starts and $\pi$
    (b) For each pair $s, a$ appearing in the episode:
            $R \leftarrow$ return following the first occurrence of $s, a$
            Append $R$ to $Returns(s,a)$
            $Q(s,a) \leftarrow$ average$(Returns(s,a))$
    (c) For each $s$ in the episode:
            $\pi(s) \leftarrow \arg\max_a Q(s,a)$
```

Action values

Policy

Returns

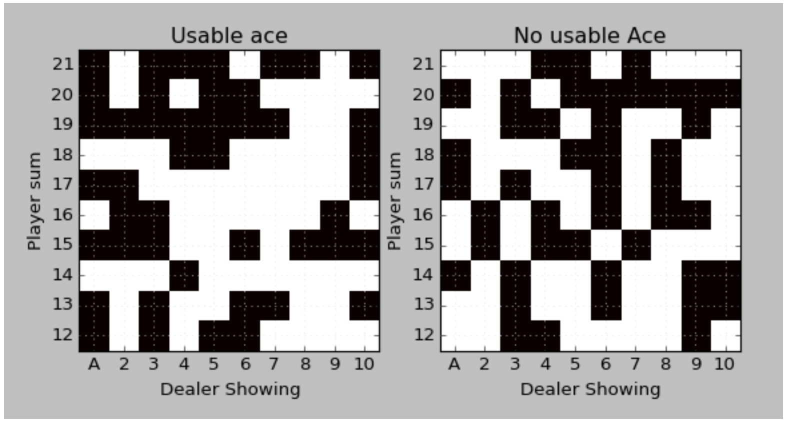
Exploration phase

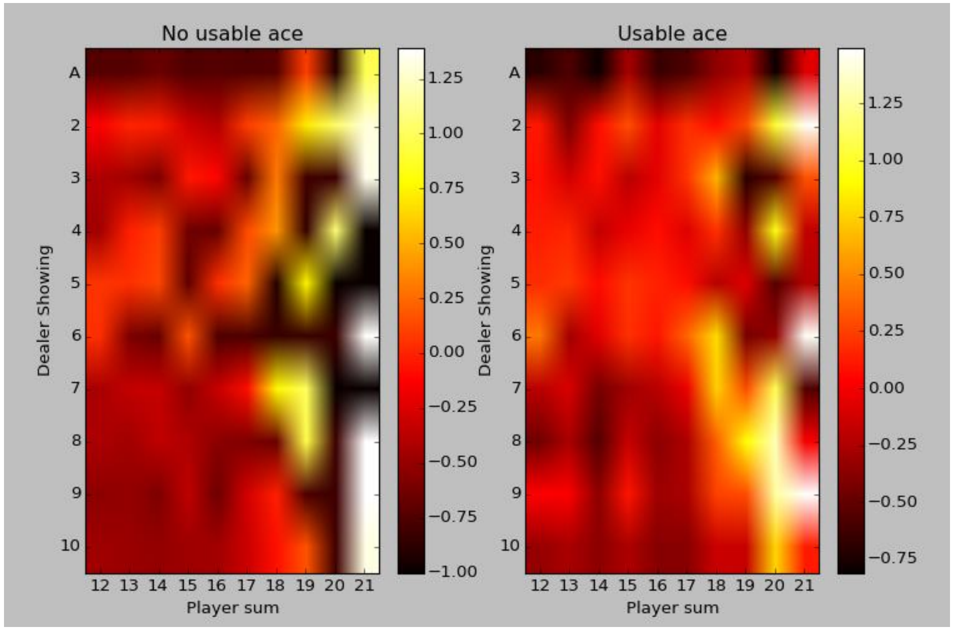
Policy evaluation

Greedy policy improvement

Recall: Q(s,a) represents the "action-value" function, and is the average reward obtained upon starting at state s, taking action a and following policy π thereafter. **Exploring starts** assumes that episodes start with state-action pairs randomly selected.

MC policy evaluation





An arbitrary policy*, π
Black: 'HIT'
White: 'STICK'

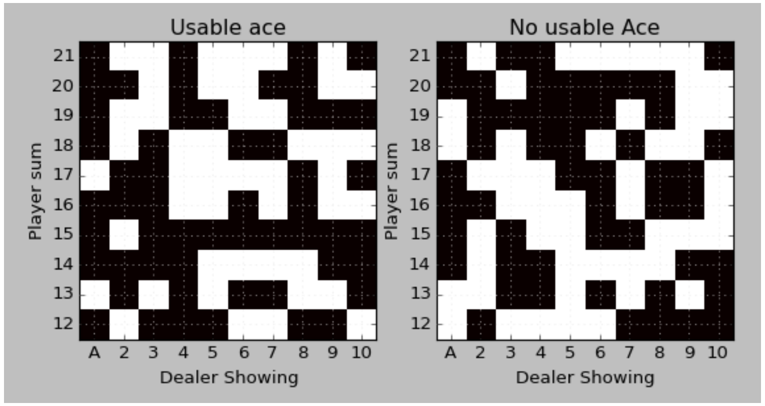
State values** (under the policy π)
Black: 'HIT'
White: 'STICK'

Poor state values suggest that the policy is sub-optimal and could be improved.

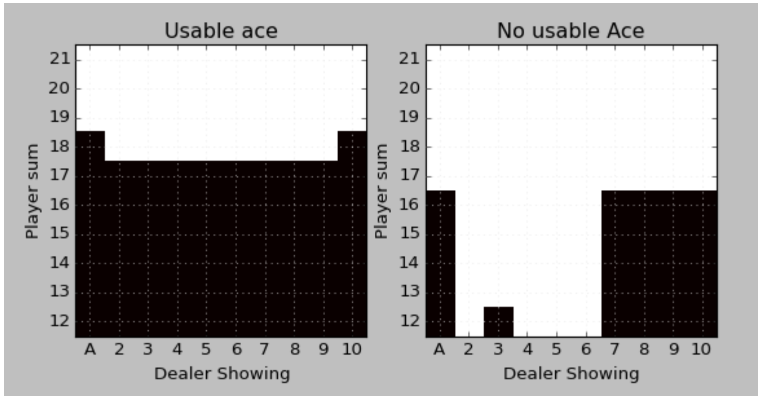
*A policy is a mapping from a state to an action.
**State values represent the average reward (winnings) from any state.

MC policy improvements

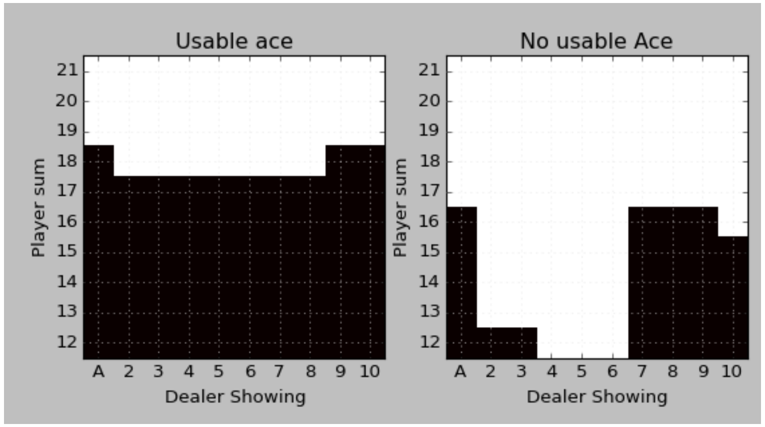
The starting policy Π(t=0)



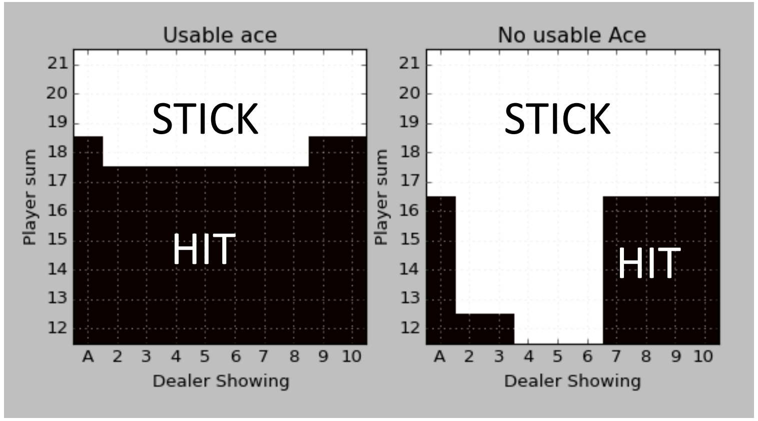
Π (after 5e5 episodes)



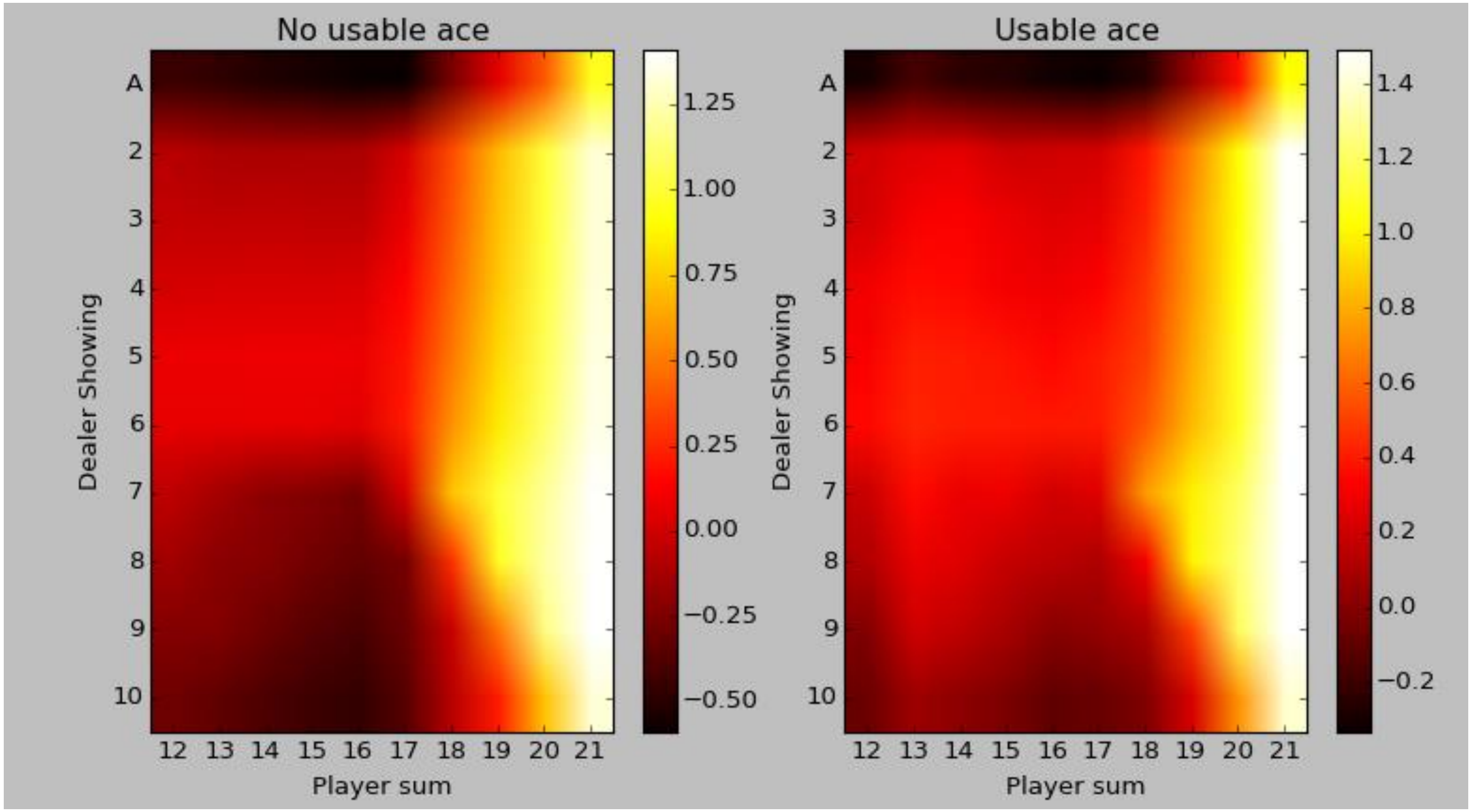
Π (after 1e6 episodes)



Π* - The optimal policy
(plotted after 5e6 episodes)



The optimal value function



Average winnings (for the optimal strategy)

- -0.425 with the payout of 3:2
  - If you play $100, you get back only 57$.
- Casinos typically use more than 1 deck to decrease these odds further
  - With 2 decks: -0.43
  - With 4 decks: -0.435
  - With 8 decks: -0.44

Avoiding exploring starts

- *Exploring starts* can sometimes be arranged in applications with simulated episodes, but are unlikely in learning from real experience. Instead, one of two general approaches can be used.
- In *on-policy* methods, the agent commits to always exploring and tries to find the best policy that still explores.
- In *off-policy* methods, the agent also explores, but learns a deterministic optimal policy that may be unrelated to the policy followed.

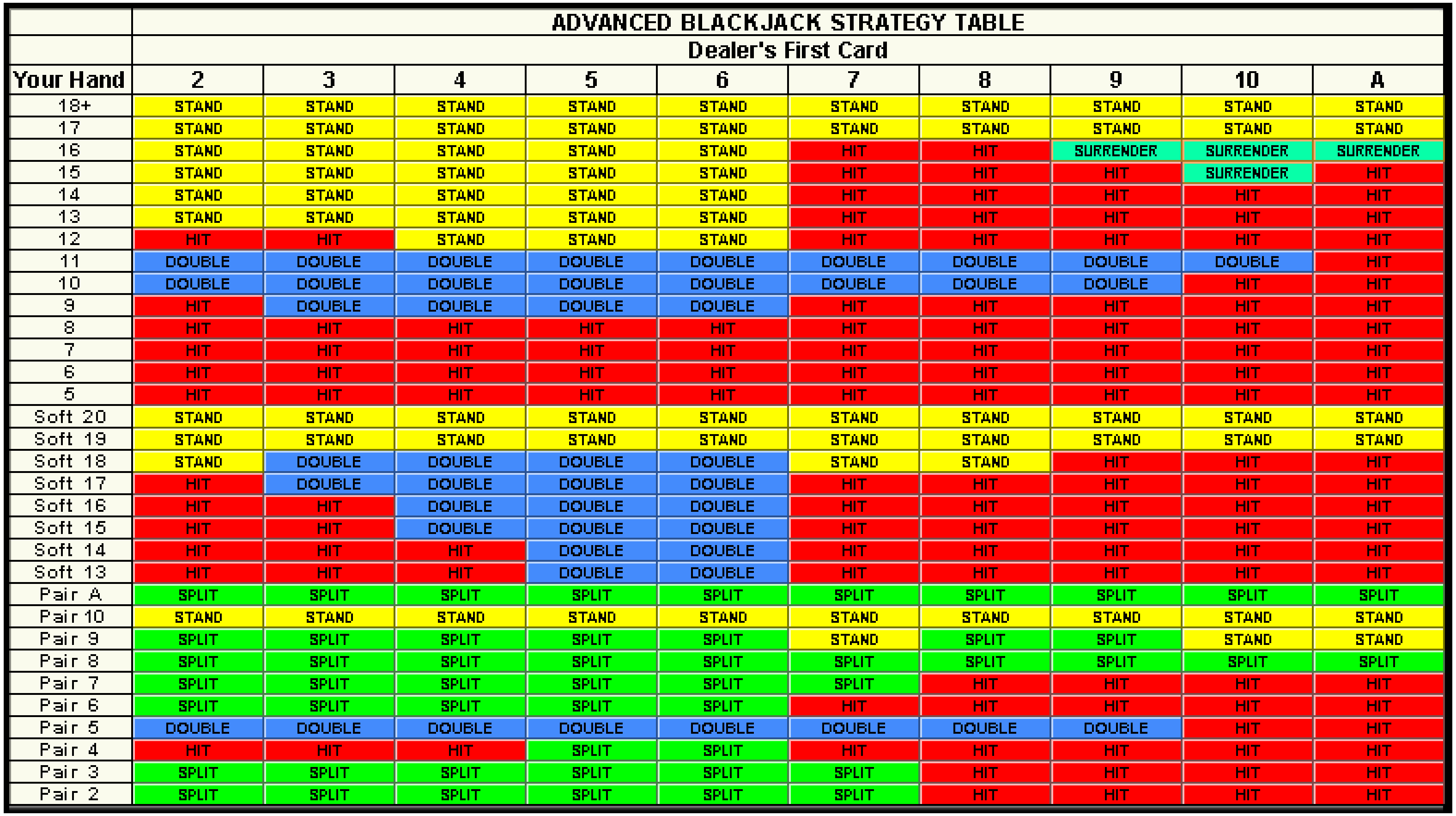
Shortcomings of MC methods

- MC methods must wait until the end of episode before return is known.
  - Hence, they work for episodic (terminating) environments.
- High error variances.
- Problematic when the number of states is too large.
- Compared to temporal difference learning, convergence will be slow.

What gives dealer the advantage?

- The house has an advantage in blackjack simply because the player has to draw first and if he busts, the player automatically loses regardless of whether the dealer would have busted or not.
- Player’s advantages over the dealer:
  - 3:2 winnings.
  - Double down, split, surrender options.
  - Dealer plays fixed strategy (stick at 17 or more).
- With the right strategy, house advantage can be reduced to -0.005, i.e., you lose 50 cents for every 100$ bet (still negative!).
- Positive advantage if you can count cards ;)

Advanced Blackjack Strategy



	ADVANCED BLACKJACK STRATEGY TABLE									
	Dealer's First Card									
Your Hand	2	3	4	5	6	7	8	9	10	A
18+	STAND	STAND	STAND	STAND	STAND	STAND	STAND	STAND	STAND	STAND
17	STAND	STAND	STAND	STAND	STAND	STAND	STAND	STAND	STAND	STAND
16	STAND	STAND	STAND	STAND	STAND	HIT	HIT	SURRENDER	SURRENDER	SURRENDER
15	STAND	STAND	STAND	STAND	STAND	HIT	HIT	HIT	SURRENDER	HIT
14	STAND	STAND	STAND	STAND	STAND	HIT	HIT	HIT	HIT	HIT
13	STAND	STAND	STAND	STAND	STAND	HIT	HIT	HIT	HIT	HIT
12	HIT	HIT	STAND	STAND	STAND	HIT	HIT	HIT	HIT	HIT
11	DOUBLE	DOUBLE	DOUBLE	DOUBLE	DOUBLE	DOUBLE	DOUBLE	DOUBLE	DOUBLE	HIT
10	DOUBLE	DOUBLE	DOUBLE	DOUBLE	DOUBLE	DOUBLE	DOUBLE	DOUBLE	HIT	HIT
9	HIT	DOUBLE	DOUBLE	DOUBLE	DOUBLE	HIT	HIT	HIT	HIT	HIT
8	HIT	HIT	HIT	HIT	HIT	HIT	HIT	HIT	HIT	HIT
7	HIT	HIT	HIT	HIT	HIT	HIT	HIT	HIT	HIT	HIT
6	HIT	HIT	HIT	HIT	HIT	HIT	HIT	HIT	HIT	HIT
5	HIT	HIT	HIT	HIT	HIT	HIT	HIT	HIT	HIT	HIT
Soft 20	STAND	STAND	STAND	STAND	STAND	STAND	STAND	STAND	STAND	STAND
Soft 19	STAND	STAND	STAND	STAND	STAND	STAND	STAND	STAND	STAND	STAND
Soft 18	STAND	DOUBLE	DOUBLE	DOUBLE	DOUBLE	STAND	STAND	HIT	HIT	HIT
Soft 17	HIT	DOUBLE	DOUBLE	DOUBLE	DOUBLE	HIT	HIT	HIT	HIT	HIT
Soft 16	HIT	HIT	DOUBLE	DOUBLE	DOUBLE	HIT	HIT	HIT	HIT	HIT
Soft 15	HIT	HIT	DOUBLE	DOUBLE	DOUBLE	HIT	HIT	HIT	HIT	HIT
Soft 14	HIT	HIT	HIT	DOUBLE	DOUBLE	HIT	HIT	HIT	HIT	HIT
Soft 13	HIT	HIT	HIT	DOUBLE	DOUBLE	HIT	HIT	HIT	HIT	HIT
Pair A	SPLIT	SPLIT	SPLIT	SPLIT	SPLIT	SPLIT	SPLIT	SPLIT	SPLIT	SPLIT
Pair 10	STAND	STAND	STAND	STAND	STAND	STAND	STAND	STAND	STAND	STAND
Pair 9	SPLIT	SPLIT	SPLIT	SPLIT	SPLIT	STAND	SPLIT	SPLIT	STAND	STAND
Pair 8	SPLIT	SPLIT	SPLIT	SPLIT	SPLIT	SPLIT	SPLIT	SPLIT	SPLIT	SPLIT
Pair 7	SPLIT	SPLIT	SPLIT	SPLIT	SPLIT	SPLIT	HIT	HIT	HIT	HIT
Pair 6	SPLIT	SPLIT	SPLIT	SPLIT	SPLIT	HIT	HIT	HIT	HIT	HIT
Pair 5	DOUBLE	DOUBLE	DOUBLE	DOUBLE	DOUBLE	DOUBLE	DOUBLE	DOUBLE	HIT	HIT
Pair 4	HIT	HIT	HIT	SPLIT	SPLIT	HIT	HIT	HIT	HIT	HIT
Pair 3	SPLIT	SPLIT	SPLIT	SPLIT	SPLIT	SPLIT	HIT	HIT	HIT	HIT
Pair 2	SPLIT	SPLIT	SPLIT	SPLIT	SPLIT	SPLIT	HIT	HIT	HIT	HIT