some notes

(MDP Trajectory)

- In model-free setting, knowing all Q*(s,a) for all state-action pairs will be more beneficial.
V(s) is basically useless without transition probs.

(Q-learning)

- One interesting consequene of such alg is that it we can train it, such that it will converge
even before the episode ends! The resulting update is -undoubtedly- still better than random q-values.
- The problem of such naive way is; using the alg in such way where we just max the Q-fn, probably
you will end up in some vicious circle, where the policy is better than random, yet not good enough
to achieve the required task over the whole episodes.
In other words, the agent is optimizing the current knowledge, yet not exploring enough to achieve
optimal results.

- Q-values learnt by Q-learning can be different from those learnt by SARSA.
- A min amount of data to update Q-values table in Q-learning will be a tuple of (s,a,r,s_)
- Q-learning does NOT find the policy that yields the highest expected sum of rewards.
It maximize the discounted rewards, NOT sum of rewards. Unless gamma=1

(Trainable Parameters)

- having x states & y actions, will yield x*y q-values

(Off-Policy Alg)

- they are those alg that can be trained both off-policy & on-policy.

* Model-Free does NOT rely on knowing the env dynamics i.e. transition probs
* Model-based alg knows rewards in davance for every state & action.
It does NOT learn by interaction with the env

* In case you know the reward & the next state, you can simply directly sample from the distribution.

(On/Off-policy)

- On-policy: agent takes actions inside the env.
- Off-policy: something else takes actions in env, the agent trains on recorded tragectories with those actions and & then tried to act optimally by itself.

- Q-learning; is more natural for off-policy setting.

In detail, Q-learning assumes that the agent will always take the optimal action even if what it observes is suboptimal. This is typical when you pre-train on some suboptimal policy & then use the agent to behave optimally.

-SARSA is more natural for on-policy.

In contrast to Q-learning, SARSA accounts for agent's exploration/errors which is useful in case agent is exploring the env by himself.

(Expected value SARSA)

- It can be applied to off-policy & still learn the optimal policy.

The problem is its expectation, where you can set its prob dist in any way. However this will greatly shape the behavior whether it would be on- or off-policy.

For example, setting prob of optimal action equal to 1, the expected value SARSA will be equal to Q-learning; therefore off-policy.

(Cross-Entropy Method)

- It can be applied to on-policy only. It does work as off-policy.