\[
\text{Sarsa } (s,a,r,s',a') \\
S_t \leftarrow R_t + \delta \hat{q}(S_{t+1}, A_{t+1}) - \hat{q}(S_t, A_t) \} \quad \text{on-policy} \\
\hat{q}(S_t, A_t) \leftarrow \hat{q}(S_t, A_t) + \alpha \delta_t \\
\text{Q-learning } (s,a,r,s') \\
S_t \leftarrow R_t + \gamma \max_a \hat{q}(S_{t+1}, a') - \hat{q}(S_t, A_t) \} \quad \text{off-policy} \\
\hat{q}(S_t, A_t) \leftarrow \hat{q}(S_t, A_t) + \alpha \delta_t \\
\text{Init } \hat{q}(A, a) \text{ arbitrarily} \\
\text{Repeat (for each episode)} \\
\quad \text{Init } S \leftarrow 0 \\
\text{Repeat (for each time step)} \\
\quad \text{Choose } a \text{ from } s \text{ using the policy derived from } \hat{q} \\
\quad \text{Take action } a, \text{ observe } r, s' \\
\quad \text{Update } \hat{q}(s, a) \leftarrow \hat{q}(s, a) + \alpha (r + \gamma \max_a \hat{q}(s', a') - \hat{q}(s, a)) \\
\text{Until } s \leftarrow s' \text{ is terminal} \\
\text{converges to } \hat{q}^\pi_0 \text{ under sampling policy } \pi_0 \\
\text{Converges to } \hat{q}^*
Course Recap

1. Definitions
   MDP
   \( s, A, P, R, d, \delta, \gamma \)
   \( v^+, q^+, v^-, q^- \)
   \( J, \pi \)

2. Dynamic Programming
   Policy evaluation + Bellman Eqn.
   Policy Improvement
   Policy Iteration
   Value Iteration
      - Bellman optimality eqn.
      - Bellman operator

3. Monte Carlo approximation

4. TD
   Same
   + Func. Approx.

5. TD(\( \lambda \))
   Same

6. Policy Gradient (\( \triangledown \))
   Theorems

Midterm
<table>
<thead>
<tr>
<th>Convergence</th>
<th>TD</th>
<th>Sarsa</th>
<th>Q-learning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tabular</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Linear</td>
<td>✔</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>General</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
</tbody>
</table>
**A-Step Returns**

- Different potential targets for value function updates.

\[ G_t^{(n)} \text{ is the } n\text{-step return} \]

\[ G_t^{(1)} = R_t + \gamma V(S_{t+1}) = G_t \]

\[ G_t^{(2)} = R_t + \gamma R_{t+1} + \gamma^2 V(S_{t+2}) \]

**Larger n gives lower bias, higher variance.**

\[ G_t^{(n)} = \sum_{k=0}^{n-1} \gamma^k R_{t+k} + \gamma^n V(S_{t+n}) \]

\[ G_t^{(MC)} = \text{Monte Carlo} = G_t \text{ or } G_t^{(MC)} \]

**Idea:** use more than one (in fact, of all): Use a weighted average, where weights sum to 1.

**\[ \lambda \text{-Returns} \]**

\[ G_t^{\lambda} = \left( \sum_{i=1}^{\infty} \lambda^i G_t^{(i)} \right) \]

where \( \lambda \in (0, 1) \)

\[ = (1-\lambda) \sum_{i=1}^{\infty} \lambda^i G_t^{(i)} \text{ if } \lambda < 1 \]

and \( G_t^{(MC)} \) otherwise

This approach is intuitively appealing to reduce variance, but not why we do it. Andy Barto says "The math works out." In particular, it works backwards for actual computation.
\[ \Omega - \text{return} \]

\[ \text{TDx... Complex... @ NIPPS 2011} \]

\[ \text{ICML 2016} \]

\[ \lambda - \text{return is MLE-estimate of } V^\pi(S_t) \text{ if:} \]

1. \( G_t^{(1)}, G_t^{(2)}, \ldots \) are independent
2. Every \( G_t^{(i)} \) is normally distributed
3. \( \text{Var}(G_t^{(i)}) = \frac{1}{\lambda_t} \) (\( \sigma^2 = \beta/\lambda_t \))
4. \( E[G_t^{(i)}] = \nu_t(S_t) \) for all \( i \)

These assumptions are false, but the rest of the math works out!

To find best \( \frac{d}{dx} + \text{set to 0} \)

\[ \sum_{i=1}^{\infty} 2(G_t^{(i)} - X)^i \lambda^i = 0 \]

\[ \sum_{i=1}^{\infty} G_t^{(i)} \lambda^i = X \sum_{i=1}^{\infty} \lambda^i \]

\[ \text{arg max}_x \sum_{i=1}^{\infty} \ln \left( \frac{1}{\sqrt{2\pi \beta}} \right) e^{-\frac{(G_t^{(i)} - X)^2}{2\beta}} \]

\[ = \text{arg max}_x \sum_{i=1}^{\infty} \ln \left( \frac{1}{\sqrt{2\pi \beta}} \right) - \frac{(G_t^{(i)} - X)^2}{2\beta} \]

\[ = \frac{\sum_{i=1}^{\infty} G_t^{(i)} \lambda^i}{\sum_{i=1}^{\infty} \lambda^i} \]

\[ = \frac{1}{\lambda} \sum_{i=1}^{\infty} G_t^{(i)} \lambda^i \]

\[ = \frac{1}{\lambda} \left( \sum_{i=1}^{\infty} G_t^{(i)} \lambda^i \right) \]

The \( \lambda \)-return.
What is the $\lambda$-return at a terminal state?

$G_T = G_{t+1}^{(T)} = \ldots$

Shows that larger $\lambda$ gives $G$’s closer to Monte Carlo.

$\lambda$-return Algorithm

Use $G_{t+1}^{(T)}$ as target. $v(S_t) \leftarrow v(S_t) + \lambda (G_T - v(S_t))$ ← "Forward view," which requires waiting until the end of the episode.