ECE276B: Planning & Learning in Robotics
Lecture 12: Model-free Prediction

Instructor:

Nikolay Atanasov: natanasov@ucsd.edu

Teaching Assistants:

Thai Duong: tduong@eng.ucsd.edu

UC San Diego

JACOBS SCHOOL OF ENGINEERING

Electrical and Computer Engineering

From Optimal Control To Reinforcement Learning

- **Stochastic Optimal Control**: MDP with known motion model $p_f(\mathbf{x}' \mid \mathbf{x}, \mathbf{u})$ and cost function $\ell(\mathbf{x}, \mathbf{u})$
  - **Model-based Prediction**: computes the value function V^π of a given policy π (policy evaluation theorem)
  - **Model-based Control**: optimizes the value function V^π to obtain an improved policy π' (policy improvement theorem)
- **Reinforcement Learning**: MDP with unknown motion model $p_f(\mathbf{x}' \mid \mathbf{x}, \mathbf{u})$ and cost function $\ell(\mathbf{x}, \mathbf{u})$ but access to samples of system transitions and incurred costs
  - **Model-free Prediction**: estimates the value function V^π of a given policy π:
    - Monte-Carlo (MC) Prediction
    - Temporal-Difference (TD) Prediction
  - **Model-free Control**: optimizes the value function V^π to obtain an improved policy π':
    - On-policy MC Control: ϵ-greedy
    - On-policy TD Control: SARSA
    - Off-policy MC Control: Importance Sampling
    - Off-policy TD Control: Q-Learning

Bellman Backup Operators

- Operators for policy-specific value functions:
  - **Policy Evaluation Backup Operator**:

$$\mathcal{B}_\pi[V](\mathbf{x}) := H[\mathbf{x}, \pi(\mathbf{x}), V(\cdot)] = \ell(\mathbf{x}, \pi(\mathbf{x})) + \gamma \mathbb{E}_{\mathbf{x}' \sim p_f(\cdot|\mathbf{x}, \pi(\mathbf{x}))}\left[V(\mathbf{x}')\right]$$

  - **Policy Q-Evaluation Backup Operator**:

$$\mathcal{B}_\pi[Q](\mathbf{x}, \mathbf{u}) := \ell(\mathbf{x}, \mathbf{u}) + \gamma \mathbb{E}_{\mathbf{x}' \sim p_f(\cdot|\mathbf{x}, \mathbf{u})}\left[Q(\mathbf{x}', \pi(\mathbf{x}'))\right]$$

- Operators for the optimal value functions:
  - **Value Iteration Backup Operator**:

$$\mathcal{B}_*[V](\mathbf{x}) := \min_{\mathbf{u} \in \mathcal{U}(\mathbf{x})} H[\mathbf{x}, \mathbf{u}, V(\cdot)] = \min_{\mathbf{u} \in \mathcal{U}(\mathbf{x})} \left\{\ell(\mathbf{x}, \mathbf{u}) + \gamma \mathbb{E}_{\mathbf{x}' \sim p_f(\cdot|\mathbf{x}, \mathbf{u})}\left[V(\mathbf{x}')\right]\right\}$$

  - **Q-Value Iteration Backup Operator**:

$$\mathcal{B}_*[Q](\mathbf{x}, \mathbf{u}) := \ell(\mathbf{x}, \mathbf{u}) + \gamma \mathbb{E}_{\mathbf{x}' \sim p_f(\cdot|\mathbf{x}, \mathbf{u})}\left[\min_{\mathbf{u}' \in \mathcal{U}(\mathbf{x}')} Q(\mathbf{x}', \mathbf{u}')\right]$$

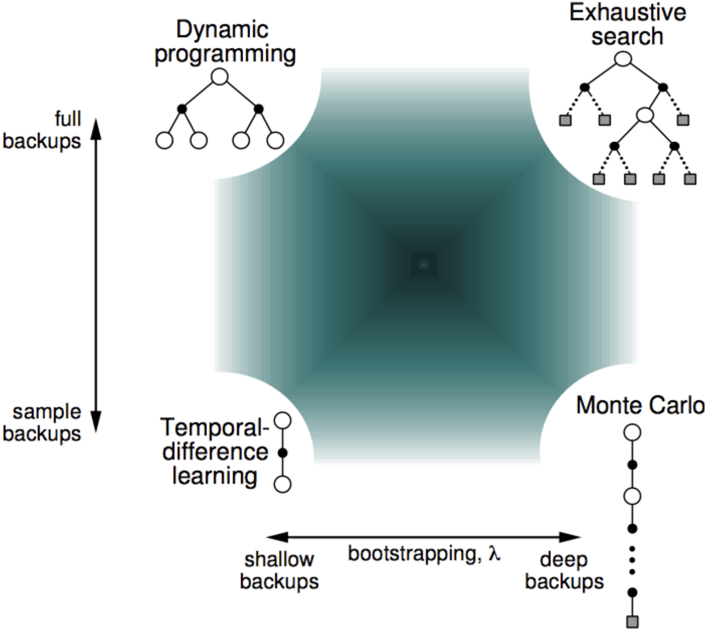
Model-free Prediction

- The main idea of model-free prediction is to approximate the Policy Evaluation backup operators $\mathcal{B}_\pi[V]$ and $\mathcal{B}_\pi[Q]$ using samples instead of computing the expectation over $\mathbf{x}'$ exactly:
  - Monte-Carlo (MC) methods:
    - The expected long-term cost can be approximated by a sample average over whole system trajectories (only applies to the First-Exit and Finite-Horizon settings)
  - Temporal-Difference (TD) methods:
    - The expected long-term cost can be approximated by a sample average over single system transitions and an estimate of the expected long-term cost at the new states (bootstrapping)

- **Sampling**: value estimates rely on samples:
  - DP does not sample
  - MC samples
  - TD samples

- **Bootstrapping**: value estimates rely on other value estimates:
  - DP bootstraps
  - MC does not bootstrap
  - TD bootstraps

Unified View of Reinforcement Learning



Monte-Carlo Policy Evaluation

- **Assumption**: MC policy evaluation applies only to the First-Exit (terminating) formulation
- **Episode**: a random sequence ρ_τ of states and controls from the start x_τ, following the system dynamics under policy π:

$$\rho_\tau := \mathbf{x}_\tau, \mathbf{u}_\tau, \mathbf{x}_{\tau+1}, \mathbf{u}_{\tau+1}, \ldots, \mathbf{x}_{T-1}, \mathbf{u}_{T-1}, \mathbf{x}_T \sim \pi$$

- **Long-term Cost**: $L_\tau(\rho_\tau) := \gamma^{T-\tau}\mathfrak{q}(\mathbf{x}_T) + \sum_{t=\tau}^{T-1} \gamma^{t-\tau}\ell(\mathbf{x}_t, \mathbf{u}_t)$
- **Goal**: approximate $V^\pi(\mathbf{x}_\tau)$ from several episodes $\rho_\tau^{(k)} \sim \pi$, $k = 1, \ldots, K$
- **MC Policy Evaluation**: uses the empirical mean of long-term costs of the episodes $\rho_\tau^{(k)}$ to approximate the value of π:

$$V^\pi(\mathbf{x}) = \mathbb{E}_{\rho\sim\pi}[L_\tau(\rho) \mid \mathbf{x}_\tau = \mathbf{x}] \approx \frac{1}{K}\sum_{k=1}^{K} L_\tau(\rho_\tau^{(k)})$$

Monte-Carlo Policy Evaluation

- **First-visit MC Policy Evaluation**:
  - for each state $\mathbf{x}$ and episode $\rho^{(k)}$, find the **first** time step t that state $\mathbf{x}$ is visited in $\rho^{(k)}$ and increment:
    - the number of visits to $\mathbf{x}$: $N(\mathbf{x}) \leftarrow N(\mathbf{x}) + 1$
    - the long-term cost starting from $\mathbf{x}$: $C(\mathbf{x}) \leftarrow C(\mathbf{x}) + L_t(\rho^{(k)})$
  - Approximate the value function of π: $V^{\pi}(\mathbf{x}) \approx \frac{C(\mathbf{x})}{N(\mathbf{x})}$
- **Every-visit MC Policy Evaluation**: same idea but the long-term costs are accumulated following **every** time step t that state $\mathbf{x}$ is visited in $\rho^{(k)}$

Monte-Carlo Policy Evaluation

Algorithm 1 First-visit MC Policy Evaluation

Initialize $V^\pi(\mathbf{x})$, $\pi(\mathbf{x})$, $C(\mathbf{x}) \leftarrow 0$, $N(\mathbf{x}) \leftarrow 0$
loop
    Generate $\rho := \mathbf{x}_0, \mathbf{u}_0, \mathbf{x}_1, \mathbf{u}_1, \ldots, \mathbf{x}_{T-1}, \mathbf{u}_{T-1}, \mathbf{x}_T$ from π
    for $\mathbf{x} \in \rho$ **do**
        $L \leftarrow$ return following first appearance of $\mathbf{x}$ in ρ
        $N(\mathbf{x}) \leftarrow N(\mathbf{x}) + 1$
        $C(\mathbf{x}) \leftarrow C(\mathbf{x}) + L$
        $V^\pi(\mathbf{x}) \leftarrow \frac{C(\mathbf{x})}{N(\mathbf{x})}$

- Every-visit MC would add to $C(\mathbf{x})$ not a single return L but the returns $\{L\}$ following all appearances of $\mathbf{x}$ in ρ

Running Sample Average

- Consider a sequence $x_1, x_2, \ldots,$ of samples from a random variable
- Usual way of computing the sample mean: $\mu_{k+1} = \frac{1}{k+1} \sum_{j=1}^{k+1} x_j$
- **Running sample average**:

$$\mu_{k+1} = \frac{1}{k+1} \sum_{j=1}^{k+1} x_j = \frac{1}{k+1} \left(x_{k+1} + \sum_{j=1}^{k} x_j \right) = \frac{1}{k+1} (x_{k+1} + k\mu_k)$$

$$= \mu_k + \frac{1}{k+1}(x_{k+1} - \mu_k)$$

- **Recency-weighted average**: update μ_k using a step-size $\alpha \neq \frac{1}{k+1}$:

$$\mu_{k+1} = \mu_k + \alpha(x_{k+1} - \mu_k) = (1-\alpha)^k x_1 + \sum_{j=1}^{k} \alpha(1-\alpha)^{k-j} x_{j+1}$$

- **Robbins-Monro Step Sizes**: convergence to the true mean is guaranteed almost surely under the following conditions:

(independence from initial conditions) $\sum_{k=1}^{\infty} \alpha_k = \infty$ $\qquad$ $\sum_{k=1}^{\infty} \alpha_k^2 < \infty$ (ensures convergence)

First-visit MC Policy Evaluation

Algorithm 2 First-visit MC Policy Evaluation

```
Initialize V^π(x), π(x)
loop
    Generate ρ := x_0, u_0, x_1, u_1, ..., x_{T-1}, u_{T-1}, x_T from π
    for x ∈ ρ do
        L ← return following first appearance of x in ρ
        V^π(x) ← V^π(x) + α(L − V^π(x))        ▷ usual choice: α := 1/(N(x)+1)
```

- The recency-weighted updates can be useful to track the value average in non-stationary problems (e.g., forgeting old episodes)

Temporal-Difference Policy Evaluation

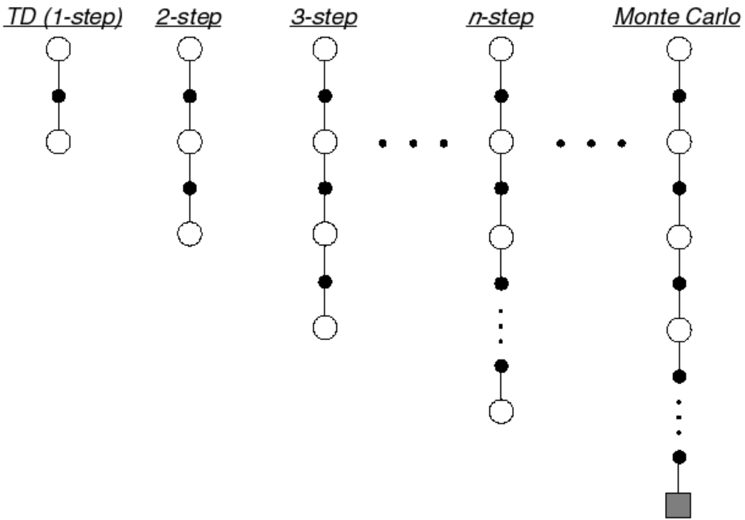
- **Bootstrapping**: the value estimate of state $\mathbf{x}$ relies on the value estimate of another state
- TD combines the sampling of MC with the bootstrapping of DP:

$$
\begin{aligned}
V^\pi(\mathbf{x}) &= \mathbb{E}_{\rho\sim\pi}[L_\tau(\rho) \mid \mathbf{x}_\tau = \mathbf{x}] \\
&= \mathbb{E}_{\rho\sim\pi}\left[\gamma^{T-\tau}\mathfrak{q}(\mathbf{x}_T) + \sum_{t=\tau}^{T-1}\gamma^{t-\tau}\ell(\mathbf{x}_t,\mathbf{u}_t) \mid \mathbf{x}_\tau = \mathbf{x}\right] \\
&= \mathbb{E}_{\rho\sim\pi}\left[\ell(\mathbf{x}_\tau,\mathbf{u}_\tau) + \gamma\left(\gamma^{T-\tau-1}\mathfrak{q}(\mathbf{x}_T) + \sum_{t=\tau+1}^{T-1}\gamma^{t-\tau-1}\ell(\mathbf{x}_t,\mathbf{u}_t)\right) \mid \mathbf{x}_\tau = \mathbf{x}\right] \\
&\underset{\text{bootstrap}}{\overset{TD(0)}{=\!=\!=}} \mathbb{E}_{\rho\sim\pi}\left[\ell(\mathbf{x}_\tau,\mathbf{u}_\tau) + \gamma V^\pi(\mathbf{x}_{\tau+1}) \mid \mathbf{x}_\tau = \mathbf{x}\right] \\
&\underset{\text{bootstrap}}{\overset{TD(n)}{=\!=\!=}} \mathbb{E}_{\rho\sim\pi}\left[\sum_{t=\tau}^{\tau+n}\gamma^{t-\tau}\ell(\mathbf{x}_t,\mathbf{u}_t) + \gamma^{n+1}V^\pi(\mathbf{x}_{\tau+n+1}) \mid \mathbf{x}_\tau = \mathbf{x}\right] \\
&\overset{MC}{\approx} \frac{1}{K}\sum_{k=1}^{K}\left[\sum_{t=\tau}^{\tau+n}\gamma^{t-\tau}\ell(\mathbf{x}_t^{(k)},\mathbf{u}_t^{(k)}) + \gamma^{n+1}V^\pi(\mathbf{x}_{\tau+n+1}^{(k)})\right]
\end{aligned}
$$

Temporal-Difference Policy Evaluation

- **Prediction**: estimate V^π from trajectory samples $\rho = \mathbf{x}_0, \mathbf{u}_0, \mathbf{x}_1, \mathbf{u}_1, \ldots, \mathbf{x}_{T-1}, \mathbf{u}_{T-1}, \mathbf{x}_T \sim \pi$
- **MC Policy Evaluation**: updates the value estimate $V^\pi(\mathbf{x}_t)$ towards the long-term cost $L_t(\rho_t)$:
$$V^\pi(\mathbf{x}_t) \leftarrow V^\pi(\mathbf{x}_t) + \alpha(L_t(\rho_t) - V^\pi(\mathbf{x}_t))$$
- **TD(0) Policy Evaluation**: updates the value estimate $V^\pi(\mathbf{x}_t)$ towards an *estimated* long-term cost $\ell(\mathbf{x}_t, \mathbf{u}_t) + \gamma V^\pi(\mathbf{x}_{t+1})$:
$$V^\pi(\mathbf{x}_t) \leftarrow V^\pi(\mathbf{x}_t) + \alpha(\ell(\mathbf{x}_t, \mathbf{u}_t) + \gamma V^\pi(\mathbf{x}_{t+1}) - V^\pi(\mathbf{x}_t))$$
- **TD(n) Policy Evaluation**: updates the value estimate $V^\pi(\mathbf{x}_t)$ towards an *estimated* long-term cost $\sum_{\tau=t}^{t+n} \gamma^{\tau-t} \ell(\mathbf{x}_\tau, \mathbf{u}_\tau) + \gamma^{n+1} V^\pi(\mathbf{x}_{t+n+1})$:
$$V^\pi(\mathbf{x}_t) \leftarrow V^\pi(\mathbf{x}_t) + \alpha \left(\sum_{\tau=t}^{t+n} \gamma^{\tau-t} \ell(\mathbf{x}_\tau, \mathbf{u}_\tau) + \gamma^{n+1} V^\pi(\mathbf{x}_{t+n+1}) - V^\pi(\mathbf{x}_t) \right)$$

TD(n) Prediction



MC and TD Errors

- **TD Error**: measures the difference between the estimated value $V^\pi(\mathbf{x}_t)$ and the better estimate $\ell(\mathbf{x}_t, \mathbf{u}_t) + \gamma V^\pi(\mathbf{x}_{t+1})$:

$$\delta_t := \ell(\mathbf{x}_t, \mathbf{u}_t) + \gamma V^\pi(\mathbf{x}_{t+1}) - V^\pi(\mathbf{x}_t)$$

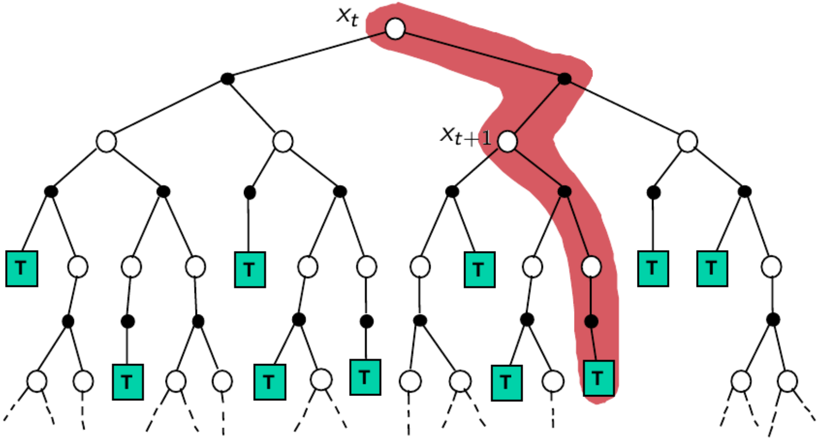
- **MC Error**: a sum of TD errors:

$$\begin{aligned} L_t(\rho_t) - V^\pi(\mathbf{x}_t) &= \ell(\mathbf{x}_t, \mathbf{u}_t) + \gamma L_{t+1}(\rho_{t+1}) - V^\pi(\mathbf{x}_t) \\ &= \delta_t + \gamma \left(L_{t+1}(\rho_{t+1}) - V^\pi(\mathbf{x}_{t+1})\right) \\ &= \delta_t + \gamma\delta_{t+1}\gamma^2 \left(L_{t+2}(\rho_{t+2}) - V^\pi(\mathbf{x}_{t+2})\right) \\ &= \sum_{n=0}^{T-t-1} \gamma^n \delta_{t+n} \end{aligned}$$

- **MC and TD converge**: $V^\pi(\mathbf{x})$ approaches the true value function of π as the number of sampled episodes $\to \infty$ as long as α_k is a Robbins-Monro sequence and $\mathcal{X}$ is finite (needed for TD convergence)

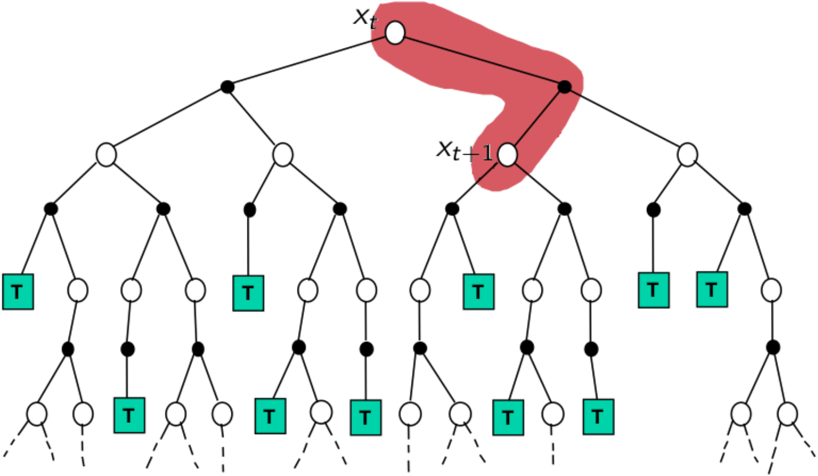
Monte-Carlo Backup

$$V^{\pi}(\mathbf{x}_t) \leftarrow V^{\pi}(\mathbf{x}_t) + \alpha(L_t(\rho_t) - V^{\pi}(\mathbf{x}_t))$$



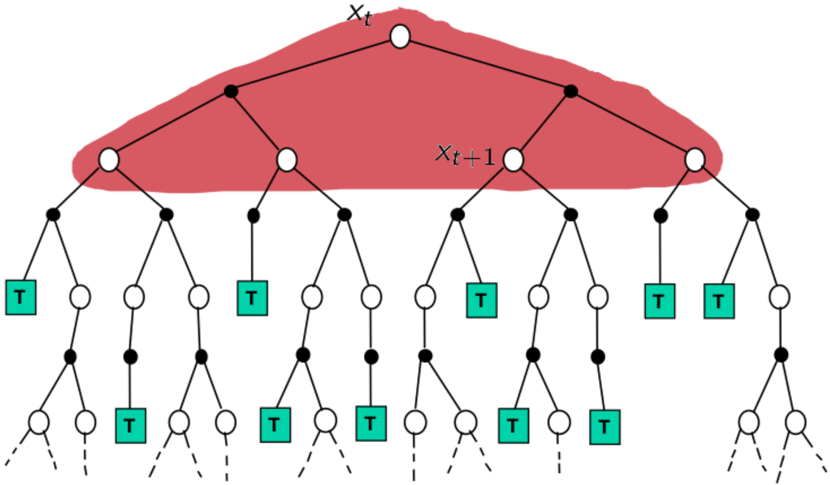
Temporal-Difference Backup

$$V^\pi(\mathbf{x}_t) \leftarrow V^\pi(\mathbf{x}_t) + \alpha(\ell(\mathbf{x}_t, \mathbf{u}_t) + \gamma V^\pi(\mathbf{x}_{t+1}) - V^\pi(\mathbf{x}_t))$$



Dynamic-Programming Backup

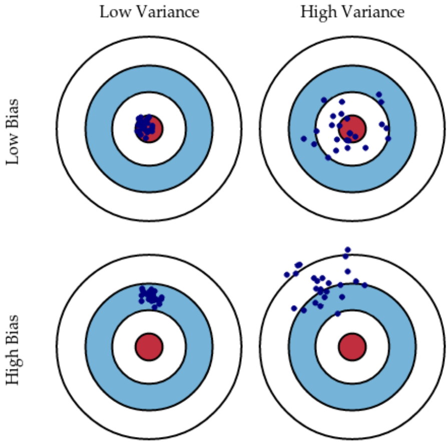
$$V^{\pi}(\mathbf{x}_t) \leftarrow \ell(\mathbf{x}_t, \mathbf{u}_t) + \gamma \mathbb{E}_{\mathbf{x}_{t+1} \sim p_f(\cdot|\mathbf{x}_t, \mathbf{u}_t)} [V^{\pi}(\mathbf{x}_{t+1})]$$



MC vs TD Policy Evaluation

- MC:
  - Must wait until the end of an episode before updating $V^\pi(\mathbf{x})$
  - The value estimates are **zero bias but high variance** (long-term cost depends on *many* random transitions)
  - Not very sensitive to initialization
  - Has good convergence properties even with function approximation (infinite state space)
- TD:
  - Can update $V^\pi(\mathbf{x})$ before knowing the complete episode and hence can learn online, after each transition, regardless of subsequent controls
  - The value estimates are **biased but low variance** (the TD(0) target depends on *one* random transition)
  - More sensitive to initialization than MC
  - May not converge with function approximation (infinite state space)

Bias-Variance Trade-off



Batch MC and TD Policy Evaluation

- **Batch setting**: given finite experience $\{\rho^{(k)}\}_{k=1}^{K}$
  - Accumulate value function updates according to MC or TD for $k = 1, \ldots, K$
  - Apply the update to the value function **only** after a complete pass through the data
  - Repeat until the value function estimate converges
- **Batch MC**: converges to V^π that best fits the observed costs:

$$V^\pi(\mathbf{x}) = \arg\min_V \sum_{k=1}^{K} \sum_{t=0}^{T_k} \left(L_t(\rho^{(k)}) - V \right)^2 \mathbb{1}\{\mathbf{x}_t^{(k)} = \mathbf{x}\}$$

- **Batch TD(0)**: converges to V^π of the maximum likelihood MDP model that best fits the observed data

$$\hat{p}_f(\mathbf{x}' \mid \mathbf{x}, \mathbf{u}) = \frac{1}{N(\mathbf{x}, \mathbf{u})} \sum_{k=1}^{K} \sum_{t=1}^{T_k} \mathbb{1}\{\mathbf{x}_t^{(k)} = \mathbf{x}, \mathbf{u}_t^{(k)} = \mathbf{u}, \mathbf{x}_{t+1}^{(k)} = \mathbf{x}'\}$$

$$\hat{\ell}(\mathbf{x}, \mathbf{u}) = \frac{1}{N(\mathbf{x}, \mathbf{u})} \sum_{k=1}^{K} \sum_{t=1}^{T_k} \mathbb{1}\{\mathbf{x}_t^{(k)} = \mathbf{x}, \mathbf{u}_t^{(k)} = \mathbf{u}\} \ell(\mathbf{x}_t^{(k)}, \mathbf{u}_t^{(k)})$$

Averaging n-Step Returns

- Define the n-step return:

$$
\begin{aligned}
L_t^{(n)}(\rho) &:= \ell(\mathbf{x}_t, \mathbf{u}_t) + \gamma \ell(\mathbf{x}_{t+1}, \mathbf{u}_{t+1}) + \ldots + \gamma^n \ell(\mathbf{x}_{t+n}, \mathbf{u}_{t+n}) + \gamma^{n+1} V^\pi(\mathbf{x}_{t+n+1}) && \color{red}{TD(n)} \\
L_t^{(0)}(\rho) &= \ell(\mathbf{x}_t, \mathbf{u}_t) + \gamma V^\pi(\mathbf{x}_{t+1}) && \color{red}{TD(0)} \\
L_t^{(1)}(\rho) &= \ell(\mathbf{x}_t, \mathbf{u}_t) + \gamma \ell(\mathbf{x}_{t+1}, \mathbf{u}_{t+1}) + \gamma^2 V^\pi(\mathbf{x}_{t+2}) && \color{red}{TD(1)} \\
&\vdots \\
L_t^{(\infty)}(\rho) &= \ell(\mathbf{x}_t, \mathbf{u}_t) + \gamma \ell(\mathbf{x}_{t+1}, \mathbf{u}_{t+1}) + \ldots + \gamma^{T-t-1} \ell(\mathbf{x}_{T-1}, \mathbf{u}_{T-1}) + \gamma^{T-t} \mathfrak{q}(\mathbf{x}_T) && \color{red}{MC}
\end{aligned}
$$

- **TD(n)**:

$$V^\pi(\mathbf{x}_t) \leftarrow V^\pi(\mathbf{x}_t) + \alpha(\color{red}{L_t^{(n)}(\rho)} - V^\pi(\mathbf{x}_t))$$

- **Averaged-return TD**: combines bootstrapping from several states:

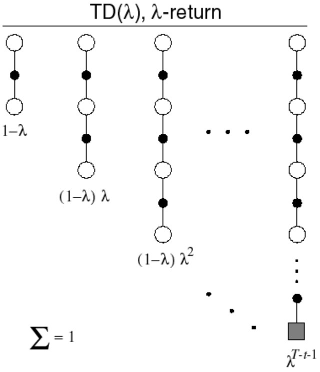
$$V^\pi(\mathbf{x}_t) \leftarrow V^\pi(\mathbf{x}_t) + \alpha \left(\color{red}{\frac{1}{2} L_t^{(2)}(\rho) + \frac{1}{2} L_t^{(4)}(\rho)} - V^\pi(\mathbf{x}_t) \right)$$

- Can we combine information from all time-steps?

Forward-view $TD(\lambda)$

- λ-**return**: combines all n-step returns:

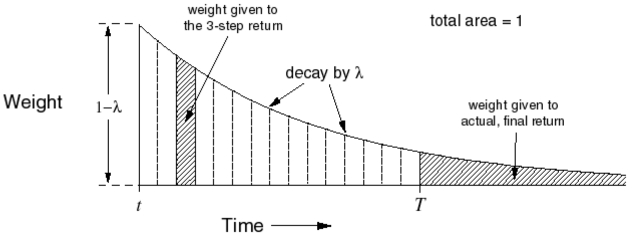
$$L_t^{\lambda}(\rho) = (1-\lambda) \sum_{n=0}^{T-t-2} \lambda^n L_t^{(n)}(\rho) + \lambda^{T-t-1} L_t^{(\infty)}$$



- **Forward-view** $TD(\lambda)$:

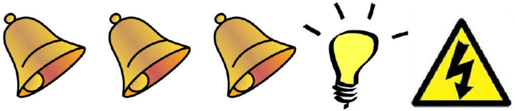
$$V^{\pi}(\mathbf{x}_t) \leftarrow V^{\pi}(\mathbf{x}_t) + \alpha \left(L_t^{\lambda}(\rho) - V^{\pi}(\mathbf{x}_t) \right)$$

- Like MC, the L_t^{λ} return can only be computed from complete episodes



Backward-view $TD(\lambda)$

- Forward-view $TD(\lambda)$ is equivalent to $TD(0)$ for $\lambda = 0$ and to every-visit MC for $\lambda = 1$
- Backward-view $TD(\lambda)$ allows online updates from incomplete episodes
- **Credit assignment problem**: did the bell or the light cause the shock?



  - **Frequency heuristic**: assigns credit to the most frequent states
  - **Recency heuristic**: assigns credit to the most recent states
  - **Eligibility trace**: combines both heuristics

$$e_t(\mathbf{x}) = \gamma\lambda e_{t-1}(\mathbf{x}) + \mathbb{1}\{\mathbf{x} = \mathbf{x}_t\}$$

- **Backward-view** $TD(\lambda)$: updates in proportion to the **TD error** δ_t and the **eligibility trace** $e_t(\mathbf{x})$:

$$V^\pi(\mathbf{x}_t) \leftarrow V^\pi(\mathbf{x}_t) + \alpha\left(\ell(\mathbf{x}_t, \mathbf{u}_t) + \gamma V^\pi(\mathbf{x}_{t+1}) - V^\pi(\mathbf{x}_t)\right) e_t(\mathbf{x}_t)$$