ECE276B: Planning & Learning in Robotics
Lecture 11: Model-free Prediction

Instructor:

Nikolay Atanasov: natanasov@ucsd.edu

Teaching Assistants:

Zhichao Li: zhl355@eng.ucsd.edu

Ehsan Zobeidi: ezobeidi@eng.ucsd.edu

Ibrahim Akbar: iakbar@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 $p_f(x' \mid x, u)$ and cost function $\ell(x, u)$
  - **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 $p_f(x' \mid x, u)$ and cost function $\ell(x, u)$ but access to examples 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:
    - On-policy MC Control: ϵ-greedy
    - On-policy TD Control: SARSA
    - Off-policy MC Control: Importance Sampling
    - Off-policy TD Control: Q-Learning

Dynamic Programming Backup Operators

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

$$\mathcal{T}_\pi[V](x) := H[x, \pi(x), V] = \ell(x, \pi(x)) + \gamma \mathbb{E}_{x' \sim p_f(\cdot|x,\pi(x))}[V(x')]$$

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

$$\mathcal{T}_\pi[Q](x,u) := \ell(x,u) + \gamma \mathbb{E}_{x' \sim p_f(\cdot|x,u)}[Q(x', \pi(x'))]$$

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

$$\mathcal{T}_*[V](x) := \min_{u \in \mathcal{U}(x)} H[x, u, V] = \min_{u \in \mathcal{U}(x)} \left\{ \ell(x,u) + \gamma \mathbb{E}_{x' \sim p_f(\cdot|x,u)}[V(x')] \right\}$$

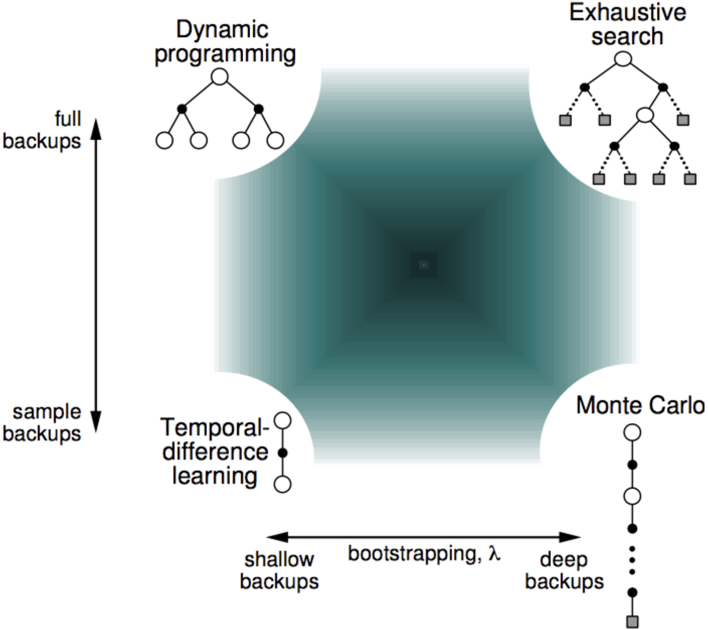
  - **Q-Value Iteration Backup Operator**:

$$\mathcal{T}_*[Q](x,u) := \ell(x,u) + \gamma \mathbb{E}_{x' \sim p_f(\cdot|x,u)} \left[\min_{u' \in \mathcal{U}(x')} Q(x', u') \right]$$

Model-free Prediction

- The main idea of model-free prediction is to approximate the Policy Evaluation backup operators $\mathcal{T}_\pi[V]$ and $\mathcal{T}_\pi[Q]$ using samples instead of computing the expectation exactly:
  - Monte-Carlo (MC) methods:
    - Expected cost can be approximated by a sample average over whole system trajectories (until termination in the SSP and final-horizon setting)
  - Temporal-Difference (TD) methods:
    - Expected cost can be approximated by a sample average over a single system transition and an estimate of the expected cost at the new state (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

- **Episode**: a random sequence ρ_t of states and controls from the start x_t, following the system dynamics to termination under policy π (SSP):

$$\rho_t := x_t, u_t, x_{t+1}, u_{t+1}, \ldots, x_{T-1}, u_{T-1}, x_T \sim \pi$$

- **Goal**: approximate $V^\pi(x_0)$ from several episodes $\rho_0^{(k)} := x_{0:T}^{(k)}, u_{0:T-1}^{(k)}$ under policy π

- Recall that the long-term cost is the sum of discounted stage costs:

$$L_t(\rho_t) = L_t(x_{t:T}, u_{t:T-1}) := \sum_{\tau=t}^{T-1} \gamma^{\tau-t} \ell(x_\tau, u_\tau) + \gamma^{T-t} \mathfrak{q}(x_T)$$

- **Monte-Carlo (MC) Policy Evaluation**: uses the empirical mean of long-term costs obtained from different episodes $\rho_t^{(k)}$ to approximate the value of π, i.e., the expected long-term cost:

$$V^\pi(x) = \mathbb{E}_{\rho\sim\pi}[L_t(\rho) \mid x_t = x] \approx \frac{1}{K}\sum_{k=1}^{K} L_t(\rho_t^{(k)})$$

First-visit Monte-Carlo Policy Evaluation

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

First-visit MC Policy Evaluation

Algorithm 1 First-visit MC Policy Evaluation

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

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

$$\left(\begin{array}{c}\text{independence from}\\ \text{initial conditions}\end{array}\right) \quad \sum_{k=1}^{\infty} \alpha_k = \infty \qquad\qquad \sum_{k=1}^{\infty} \alpha_k^2 < \infty \ \ \text{(ensure convergence)}$$

First-visit MC Policy Evaluation

Algorithm 2 First-visit MC Policy Evaluation

```
Initialize V^π(x), π(x)
loop
    Generate ρ := (x_{0:T}, u_{0:T−1}) 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 x relies on the value estimate of another state
- TD combines the sampling of MC with the bootstrapping of DP:

$$
\begin{aligned}
V^{\pi}(x) &= \mathbb{E}_{\rho\sim\pi}[L_t(\rho) \mid x_t = x] \\
&\overset{MC}{=\!=} \mathbb{E}_{\rho\sim\pi}\left[\sum_{\tau=t}^{T-1} \gamma^{\tau-t}\ell(x_\tau, u_\tau) + \gamma^{T-t}\mathfrak{q}(x_T) \mid x_t = x\right] \\
&= \mathbb{E}_{\rho\sim\pi}\left[\ell(x_t, u_t) + \gamma\left(\sum_{\tau=t+1}^{T-1} \gamma^{\tau-t-1}\ell(x_\tau, u_\tau) + \gamma^{T-t-1}\mathfrak{q}(x_T)\right) \mid x_t = x\right] \\
&\underset{\text{bootstrap}}{\overset{TD(0)}{=\!=\!=}} \mathbb{E}_{\rho\sim\pi}\left[\ell(x_t, u_t) + \gamma V^{\pi}(x_{t+1}) \mid x_t = x\right] \\
&\underset{\text{bootstrap}}{\overset{TD(n)}{=\!=\!=}} \mathbb{E}_{\rho\sim\pi}\left[\sum_{\tau=t}^{t+n} \gamma^{\tau-t}\ell(x_\tau, u_\tau) + \gamma^{n+1}V^{\pi}(x_{t+n+1}) \mid x_t = x\right]
\end{aligned}
$$

Temporal-Difference Policy Evaluation

- **Prediction**: estimate V^π from trajectory samples $\rho = x_{0:T}, u_{0:T-1} \sim \pi$
- **MC Policy Evaluation**: updates the value estimate $V^\pi(x_t)$ towards the long-term cost $L_t(x_{t:T}, u_{t:T-1})$:

$$V^\pi(x_t) \leftarrow V^\pi(x_t) + \alpha(L_t(x_{t:T}, u_{t:T-1}) - V^\pi(x_t))$$

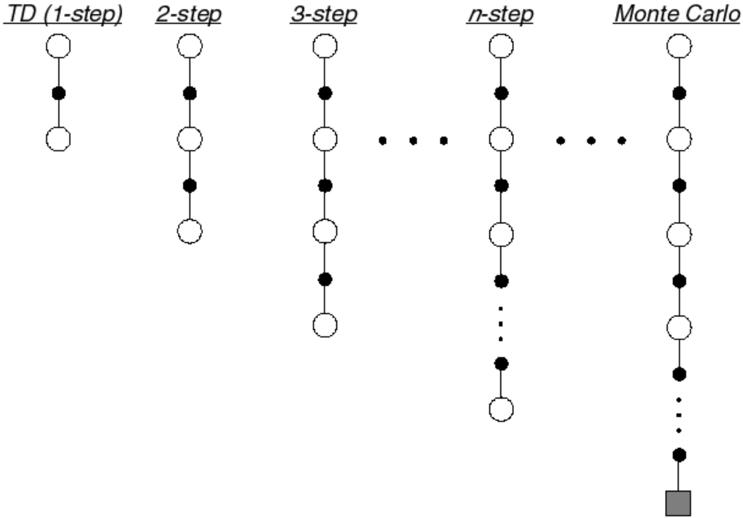
- **TD(0) Policy Evaluation**: updates the value estimate $V^\pi(x_t)$ towards an *estimated* long-term cost $\ell(x_t, u_t) + \gamma V^\pi(x_{t+1})$:

$$V^\pi(x_t) \leftarrow V^\pi(x_t) + \alpha(\ell(x_t, u_t) + \gamma V^\pi(x_{t+1}) - V^\pi(x_t))$$

- **TD(n) Policy Evaluation**: updates the value estimate $V^\pi(x_t)$ towards an *estimated* long-term cost $\sum_{\tau=t}^{t+n} \gamma^{\tau-t}\ell(x_\tau, u_\tau) + \gamma^{n+1} V^\pi(x_{t+n+1})$:

$$V^\pi(x_t) \leftarrow V^\pi(x_t) + \alpha\left(\sum_{\tau=t}^{t+n} \gamma^{\tau-t}\ell(x_\tau, u_\tau) + \gamma^{n+1} V^\pi(x_{t+n+1}) - V^\pi(x_t)\right)$$

TD(n) Prediction



MC and TD Errors

- **TD Error**: measures the difference between the estimated value $V^\pi(x_t)$ and the better estimate $\ell(x_t, u_t) + \gamma V^\pi(x_{t+1})$:

$$\delta_t := \ell(x_t, u_t) + \gamma V^\pi(x_{t+1}) - V^\pi(x_t)$$

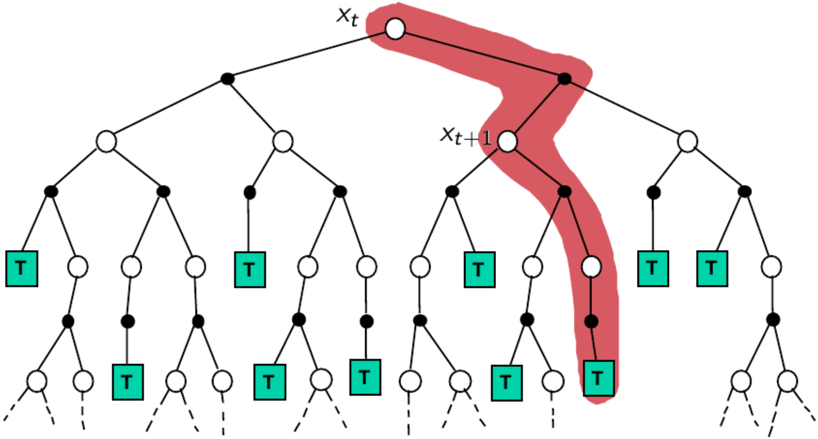
- **MC Error**: a sum of TD errors:

$$\begin{aligned}
L_t(x_{t:T}, u_{t:T-1}) - V^\pi(x_t) &= \ell(x_t, u_t) + \gamma L_{t+1}(x_{t+1:T}, u_{t+1:T-1}) - V^\pi(x_t) \\
&= \delta_t + \gamma \left(L_{t+1}(x_{t+1:T}, u_{t+1:T-1}) - V^\pi(x_{t+1}) \right) \\
&= \delta_t + \gamma \delta_{t+1} \gamma^2 \left(L_{t+2}(x_{t+2:T}, u_{t+2:T-1}) - V^\pi(x_{t+2}) \right) \\
&= \sum_{n=0}^{T-t-1} \gamma^n \delta_{t+n}
\end{aligned}$$

- **MC and TD converge**: $V^\pi(x)$ approaches the true value 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)

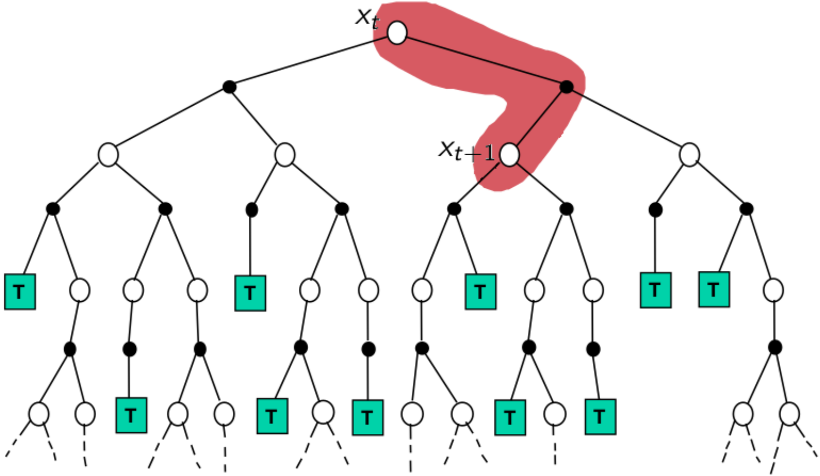
Monte-Carlo Backup

$$V^\pi(x_t) \leftarrow V^\pi(x_t) + \alpha(L_t(x_{t:T}, u_{t:T-1}) - V^\pi(x_t))$$



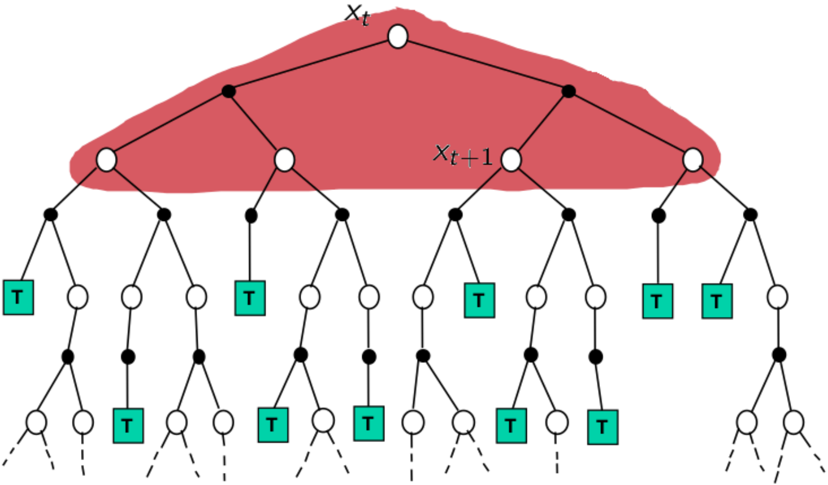
Temporal-Difference Backup

$$V^{\pi}(x_t) \leftarrow V^{\pi}(x_t) + \alpha(\ell(x_t, u_t) + \gamma V^{\pi}(x_{t+1}) - V^{\pi}(x_t))$$



Dynamic-Programming Backup

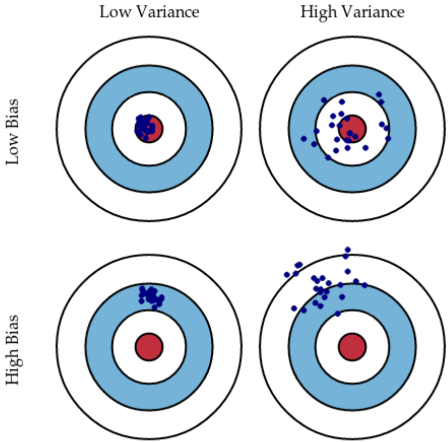
$$V^\pi(x_t) \leftarrow \ell(x_t, u_t) + \gamma \mathbb{E}_{x_{t+1} \sim p_f(\cdot|x_t, u_t)} [V^\pi(x_{t+1})]$$



MC vs TD Policy Evaluation

- MC:
  - Must wait until the end of an episode before updating $V^{\pi}(x)$
  - 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 (i.e., non-tabular setting)
- TD:
  - Can update $V^{\pi}(x)$ 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** (TD(0) target depends on *one* random transition)
  - More sensitive to initialization than MC
  - May not converge with function approximation (i.e., non-tabular setting)

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:

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

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

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

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

Averaging n-Step Returns

- Define the n-step return:

$$
\begin{aligned}
L_t^{(n)}(\rho) &:= \ell(x_t, u_t) + \gamma \ell(x_{t+1}, u_{t+1}) + \ldots + \gamma^n \ell(x_{t+n}, u_{t+n}) + \gamma^{n+1} V^\pi(x_{t+n+1}) \\
L_t^{(0)}(\rho) &= \ell(x_t, u_t) + \gamma V^\pi(x_{t+1}) & (TD(0)) \\
L_t^{(1)}(\rho) &= \ell(x_t, u_t) + \gamma \ell(x_{t+1}, u_{t+1}) + \gamma^2 V^\pi(x_{t+2}) \\
&\vdots \\
L_t^{(\infty)}(\rho) &= \ell(x_t, u_t) + \gamma \ell(x_{t+1}, u_{t+1}) + \ldots + \gamma^{T-t-1} \ell(x_{T-1}, u_{T-1}) + \gamma^{T-t} \mathfrak{q}(x_T) & (MC)
\end{aligned}
$$

- **TD(n)**:

$$V^\pi(x_t) \leftarrow V^\pi(x_t) + \alpha(L_t^{(n)}(\rho) - V^\pi(x_t))$$

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

$$V^\pi(x_t) \leftarrow V^\pi(x_t) + \alpha\left(\frac{1}{2} L_t^{(2)}(\rho) + \frac{1}{2} L_t^{(4)}(\rho) - V^\pi(x_t)\right)$$

- Can we combine information from all time-steps?

Forward-view $TD(\lambda)$

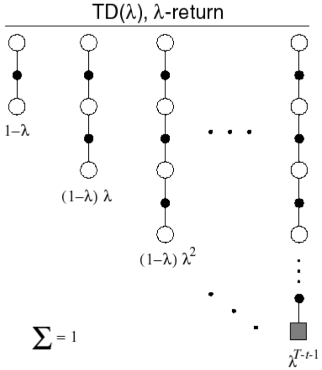
- λ-**return**: combines all n-step returns:

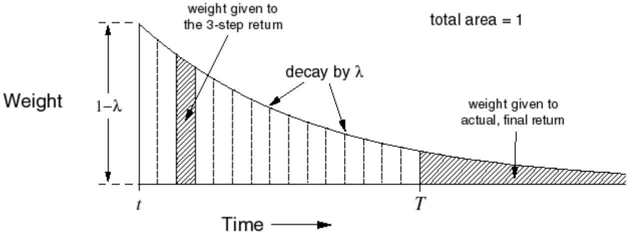
$$L_t^{\lambda}(\rho) = (1-\lambda)\sum_{n=0}^{\infty}\lambda^n L_t^{(n)}(\rho)$$

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

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

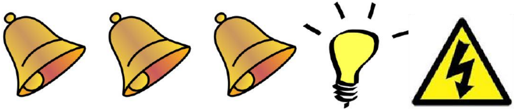
- Like MC, the L_t^{λ} return can only be computed from complete episodes





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

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

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

$$V^\pi(x_t) \leftarrow V^\pi(x_t) + \alpha \left(\ell(x_t, u_t) + \gamma V^\pi(x_{t+1}) - V^\pi(x_t) \right) e_t(x_t)$$