ECE276B: Planning & Learning in Robotics
Lecture 15: Model-free Prediction

Lecturer:

Nikolay Atanasov: natanasov@ucsd.edu

Teaching Assistants:

Tianyu Wang: tiw161@eng.ucsd.edu

Yongxi Lu: yol070@eng.ucsd.edu

UC San Diego

JACOBS SCHOOL OF ENGINEERING

Electrical and Computer Engineering

From Optimal Control To Reinforcement Learning

- **Optimal Control**:
  - **Discrete-time**: dynamic programming, search-based planning, sampling-based planning, model-based policy evaluation and improvement via generalized policy iteration
  - **Continuous-time**: Hamiltonian-Jacobi-Bellman partial differential equation (HJB PDE), Pontryagin's Minimum Principle (PMP), Linear Quadratic Regulator (LQR), Linear Quadratic Gaussian (LQG)
- **Reinforcement Learning**: no knowledge of the Markov Decision Process (MDP) motion model $p_f(x' \mid x, u)$ or cost function $g(x, u)$ but access to examples of system transitions and incurred costs
  - **Model-free Prediction**: estimate the value function of an MDP with an unknown transition model:
    - Monte-Carlo (MC) Prediction
    - Temporal-Difference (TD) Prediction
  - **Model-free Control**: optimize the value function of an MDP with an unknown transition model:
    - On-policy MC Control: ϵ-greedy
    - On-policy TD Control: SARSA
    - Off-policy MC Control: Importance Sampling
    - Off-policy TD Control: Q-Learning

Value Function

- **Value Function**: the expected long-term cost of following policy π starting at state x:

$$\begin{aligned} V^\pi(x) :=& \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t g(x_t, \pi(x_t)) \middle| x_0 = x\right] \\ =& g(x, \pi(x)) + \gamma \mathbb{E}\left[\sum_{t=1}^{\infty} \gamma^{t-1} g(x_t, \pi(x_t)) \middle| x_0 = x\right] \\ =& g(x, \pi(x)) + \gamma \mathbb{E}_{x' \sim p_f(\cdot|x,\pi(x))}\left[V^\pi(x')\right] \end{aligned}$$

- **Value Iteration**: computes the optimal value function

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

Action-Value (Q) Function

- **Q Function**: the expected long-term cost of taking action u in state x and following policy π afterwards:

$$
\begin{aligned}
Q^\pi(x,u) :=& g(x,u) + \mathbb{E}\left[\sum_{t=1}^{\infty} \gamma^t g(x_t, \pi(x_t)) \middle| x_0 = x\right] \\
=& g(x,u) + \gamma \mathbb{E}_{x' \sim p_f(\cdot|x,u)}\left[V^\pi(x')\right] \\
=& g(x,u) + \gamma \mathbb{E}_{x' \sim p_f(\cdot|x,u)}\left[Q^\pi(x', \pi(x'))\right]
\end{aligned}
$$

- **Q-Value Iteration**: computes the optimal Q function

$$
\begin{aligned}
Q^*(x,u) := \min_\pi Q^\pi(x,u) =& g(x,u) + \gamma \mathbb{E}_{x' \sim p_f(\cdot|x,u)}\left[\min_\pi V^\pi(x')\right] \\
=& g(x,u) + \gamma \mathbb{E}_{x' \sim p_f(\cdot|x,u)}\left[V^*(x')\right] \\
=& g(x,u) + \gamma \mathbb{E}_{x' \sim p_f(\cdot|x,u)}\left[\min_{u' \in \mathcal{U}(x')} Q^*(x', u')\right]
\end{aligned}
$$

Dynamic Programming Backup Operators

- Operators for policy-specific cost-to-go:
  - **Policy Evaluation Backup Operator**:

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

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

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

- Operators for the optimal cost-to-go:
  - **Value Iteration Backup Operator**:

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

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

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

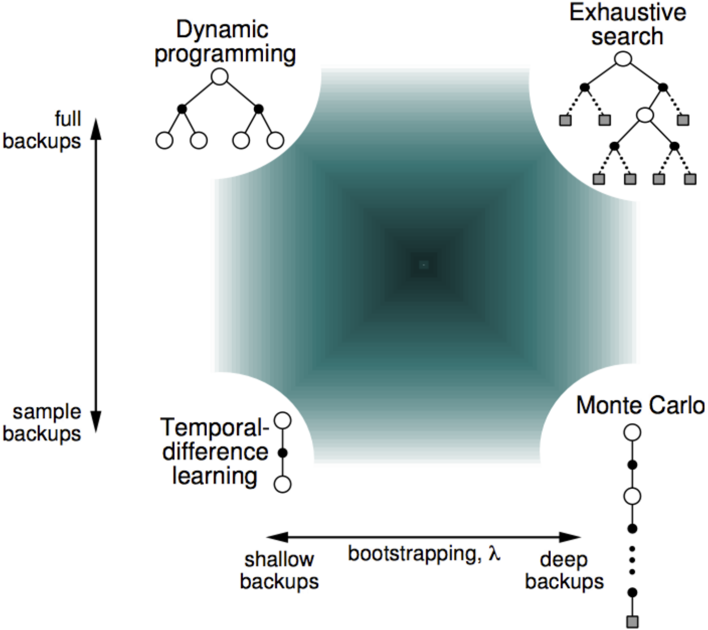
Model-free Prediction

- 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
    - Only applies to terminating problems: finite-horizon and SSP
  - 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
    - Applies to both finite- and infinite-horizon problems due to bootstrapping

- **Sampling**: cost-to-go estimates rely on samples:
  - DP does not sample
  - MC samples
  - TD samples

- **Bootstrapping**: cost-to-go estimates rely on other cost-to-go estimates:
  - DP bootstraps
  - MC does not bootstrap
  - TD bootstraps

Unified View of Reinforcement Learning



Monte-Carlo Policy Evaluation

- Applies only to terminating infinite-horizon problems
- **Episode**: a (random) sequence of states and controls from start to termination under under policy π:

$$\rho := x_0, u_0, x_1, u_1, \ldots, x_{T-1}, u_{T-1}, x_T \sim \pi$$

- **Goal**: approximate $J^\pi(x_0)$ from several episodes $\rho^{(k)} := x_{0:T}^{(k)}, u_{0:T-1}^{(k)}$ under policy π
- Recall that the long-term cost is the sum of discounted stage costs:

$$G_t(x_{t:T}, u_{t:T-1}) := \sum_{\tau=t}^{T-1} \gamma^{\tau-t} g(x_\tau, u_\tau) + \gamma^{T-t} g_T(x_T)$$

- **Monte-Carlo (MC) Policy Evaluation**: uses the empirical mean of long-term costs obtained from different episodes to approximate the cost-to-go of π, i.e., the expected long-term cost:

$$J^\pi(x) = \mathbb{E}[G_t(\rho) \mid x_t = x, \rho \sim \pi] \approx V^\pi(x) := \frac{1}{K} \sum_{k=1}^{K} G_t(\rho^{(k)})$$

First-visit Monte-Carlo Policy Evaluation

- **Prediction**: estimate $J^\pi(x)$ 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: $C(x) \leftarrow C(x) + G_t(\rho^{(k)})$
- Approximate cost-to-go: $J^\pi(x) \approx \frac{C(x)}{N(x)}$
- **Every-visit MC Policy Evaluation**: same idea but the long-term costs are averaged 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^π(x), π(x), C(x) ← ∅
loop
    Generate ρ := (x_{0:T}, u_{0:T−1}) from π
    for x ∈ ρ do
        G ← return following first appearance of x in ρ
        C(x) ← C(x) ∪ {G}
        V^π(x) ← avg(C(x))
```

- Every-visit MC would append to $C(x)$ not a single return G but the returns $\{G\}$ 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**:

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

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

- The recency-weighted updates can be useful to track the cost-to-go average in non-stationary problems (i.e., forget old episodes)

Temporal-Difference Policy Evaluation

- Applies to both terminating and non-terminating settings (incomplete episodes) since it relies on bootstrapping
- **Bootstrapping**: the cost-to-go estimate of state x relies on the cost-to-go estimate of another state
- TD combines the sampling of MC with the bootstrapping of DP:

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

Temporal-Difference Policy Evaluation

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

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

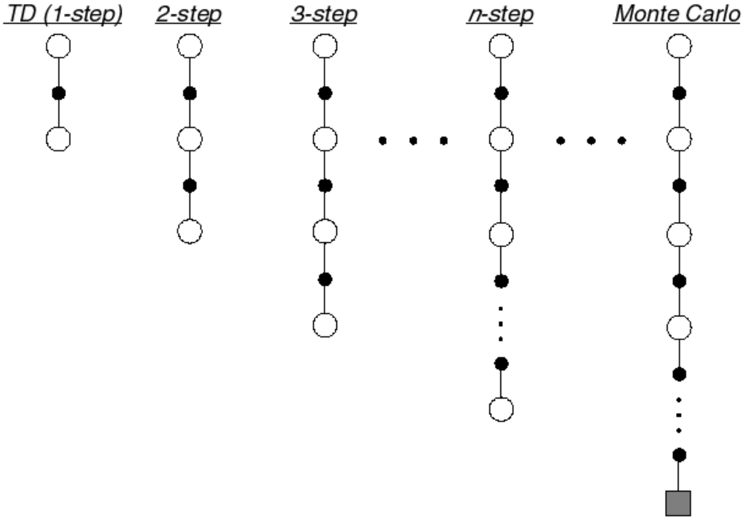
- **TD(0) Policy Evaluation**: updates the cost-to-go estimate $V^\pi(x_t)$ towards an *estimated* long-term cost $g(x_t, u_t) + \gamma V^\pi(x_{t+1})$:

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

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

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

TD(n) Prediction



MC and TD Errors

- **TD Target**: $G_t^{(0)}(\rho) := g(x_t, u_t) + \gamma V^\pi(x_{t+1})$
- **TD Error**: measures the difference between the estimated value $V^\pi(x_t)$ and the better estimate $g(x_t, u_t) + \gamma V^\pi(x_{t+1})$:

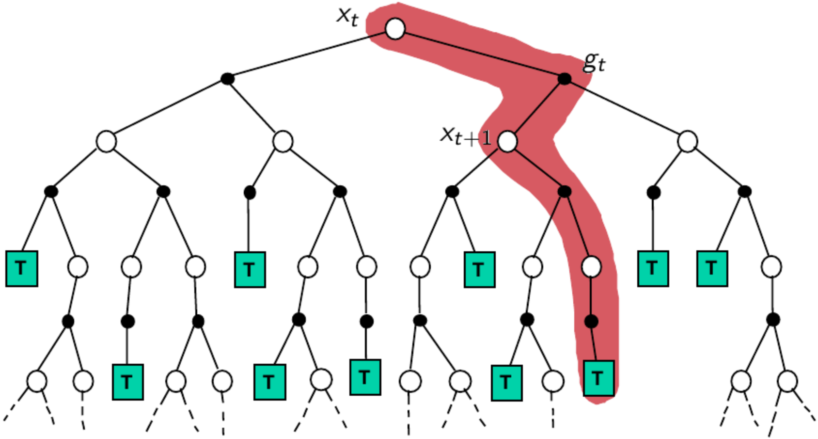
$$\delta_t := g(x_t, u_t) + \gamma V^\pi(x_{t+1}) - V^\pi(x_t)$$

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

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

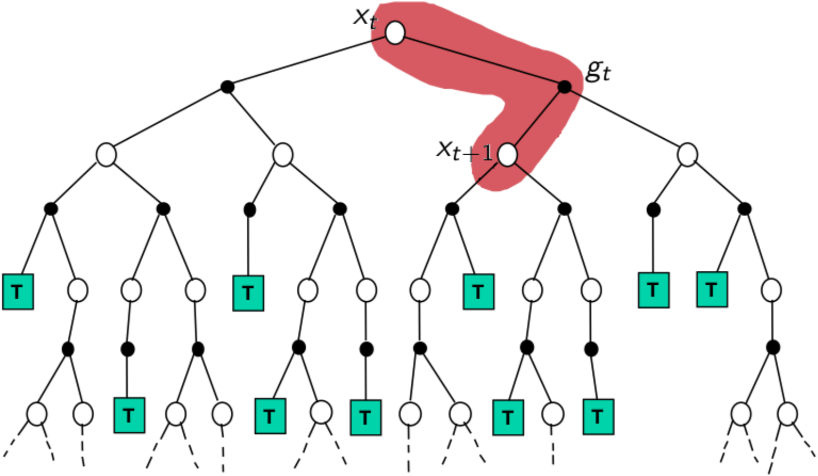
Monte-Carlo Backup

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



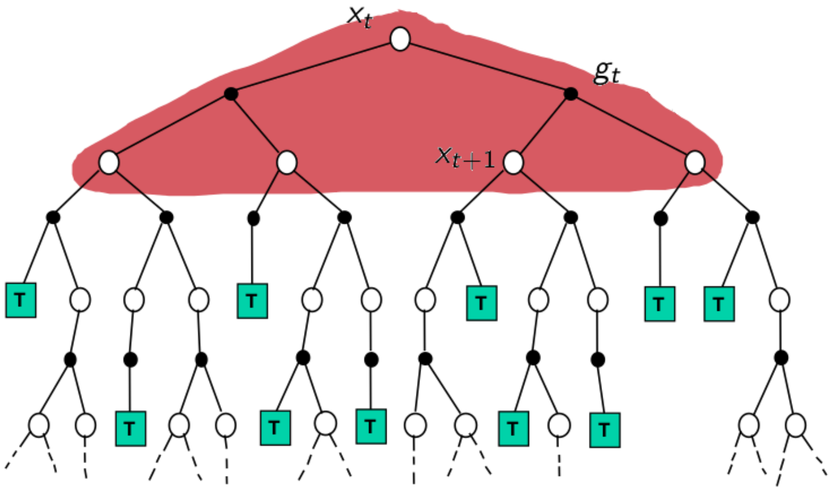
Temporal-Difference Backup

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



Dynamic-Programming Backup

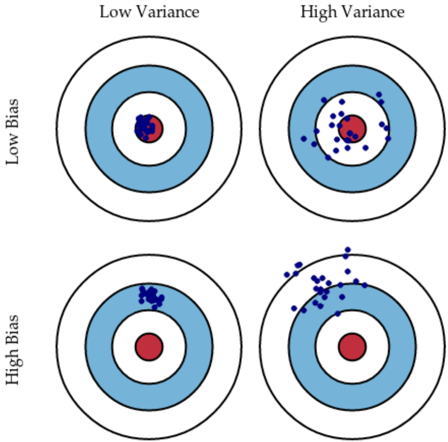
$$V^\pi(x_t) \leftarrow g(x_t, u_t) + \gamma \mathbb{E}_{x_{t+1} \sim p_f(\cdot|x_t, u_t)} [V^\pi(x_{t+1})]$$



MC vs TD Policy Evaluation

- MC:
  - Must wait until the end of an episode before updating $V^\pi(x)$
  - Works only for episodic (terminating) problems
  - The MC 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
  - Works in continuing (non-terminating) problems
  - The TD 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



Bias-Variance Decomposition

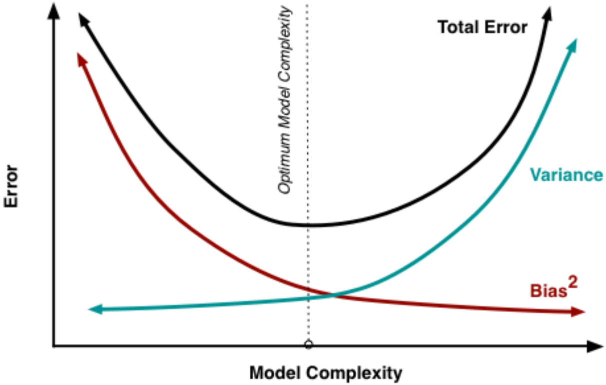
- Given **iid** data $D = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$, a new independent sample $(\mathbf{x}, y)$, and a regression model $h(\mathbf{x}, D) = \hat{y}$, the expected squared error of h is:

$$\mathbb{E}_{(\mathbf{x},y),D}(h(\mathbf{x}, D) - y)^2 = \underbrace{\mathbb{E}_{\mathbf{x},D}(h(\mathbf{x}, D) - \bar{h}(\mathbf{x}))^2}_{\textbf{Variance (overfitting)}} + \underbrace{\mathbb{E}_{\mathbf{x}}(\bar{h}(\mathbf{x}) - \bar{y}(\mathbf{x}))^2}_{\textbf{Bias}^2\textbf{ (underfitting)}}$$
$$+ \underbrace{\mathbb{E}_{(\mathbf{x},y)}(\bar{y}(\mathbf{x}) - y)^2}_{\textbf{Noise}}$$

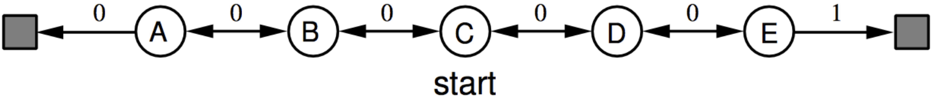
where $\bar{h}(\mathbf{x}) := \mathbb{E}_D h(\mathbf{x}, D)$ and $\bar{y}(\mathbf{x}) := \mathbb{E}[y \mid \mathbf{x}]$.

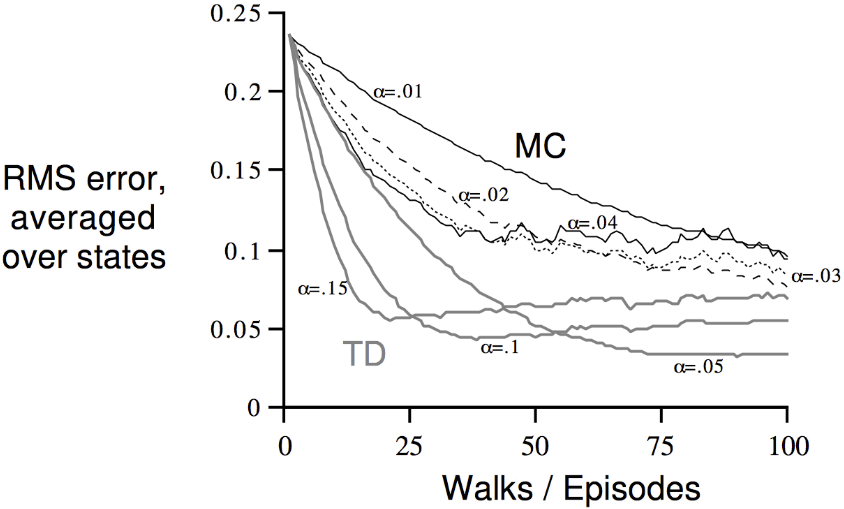
- Bias is independent of n and decreases with model complexity
- Variance decreases with n and increases with model complexity
- **Occum's Razor** the best h from a family of good models $\mathcal{H}$ is the one with the lowest complexity

Bias-Variance Decomposition



Example: Random Walk





Batch MC and TD Policy Evaluation

- **MC and TD converge**: $V^\pi(x) \to J^\pi(x)$ as the number of sampled episodes $\to \infty$ as long as α_k is a Robbins-Monro sequence
- **Batch setting**: given finite experience $\{\rho^{(k)}\}_{k=1}^{K}$, repeatedly sample $k \in [1, K]$ and apply MC or TD to episode k
- **Batch MC**: converges to V^π that best fits the observed costs:

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

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

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

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

Averaging n-Step Returns

- Define the n-step return:

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

- $TD(n)$:

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

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

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

- Can we combine information from all time-steps?

λ-Return and Forward-view $TD(\lambda)$

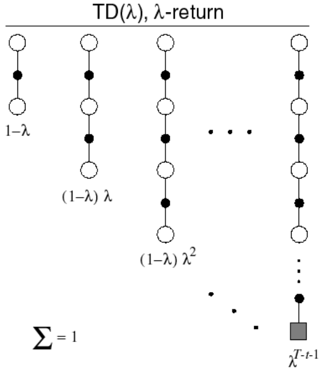
- λ-**return**: combines all n-step returns:

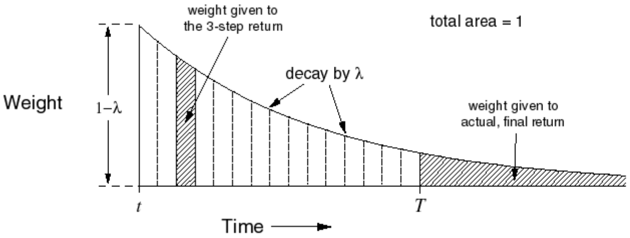
$$G_t^\lambda(\rho) = (1-\lambda)\sum_{n=0}^{\infty} \lambda^n G_t^{(n)}(\rho)$$

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

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

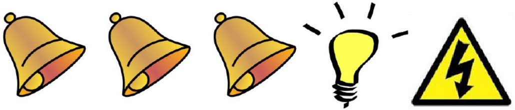
- Like MC, the G_t^λ return can only be computed from complete episodes





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 traces**: combine 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** $e_t(x)$:

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

Next Quarter (Really Soon)

- **ECE272B: Dynamical Systems under Uncertainty**
  - More rigorous treatment of Markov Chain and MDP theory
  - A more careful look at the partially observable case
  - Check out Piazza for details and a tentative syllabus
- **ECE276C: Robot Reinforcement Learning**
  - We only touched the surface of reinforcement learning; ECE276C will continue the story
  - Function Approximation, Policy Gradients, Deep neural networks in Reinforcement Learning
  - More details: `https://sites.google.com/site/mikeyip1/teaching/ece276c`