Giải thích chung

- Trong policy evaluation (ước lượng hàm giá trị của một chính sách cố định), ta muốn tìm tham số w sao cho hàm xấp xỉ $\hat{v}(s;w)$ gần với giá trị thật $v_\pi(s)$. Khi không gian trạng thái lớn, dùng function approximation (ví dụ linear hoặc neural network) để xấp xỉ thay vì lưu bảng.
- Temporal Difference (TD) learning cập nhật ước lượng giá trị "liên tục" từng bước thời gian bằng cách bootstrap (dùng ước lượng kế tiếp làm target), thay vì chờ đến hết tập episode như Monte Carlo. Điều này giúp sample efficiency cao và khả năng áp dụng trong môi trường không có ranh giới rõ rệt giữa các episode. amreis.github.io web.eecs.umich.edu .

---

2. Gradient Monte Carlo (GM) Update

Slide tóm tắt:

- GM cập nhật giá trị dựa trên sample return $G_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1}$. Công thức update dạng gradient descent giảm thiểu sai số bình phương giữa $\hat{v}(s_t;w)$ và G_t.
- Vì G_t là unbiased estimate của giá trị thật $v_\pi(s_t)$, nên Gradient Monte Carlo áp dụng đúng gradient descent trên Mean Squared Value Error (MSVE) và hội tụ (đến một local minimum nếu function approximation phi tuyến; với linear cũng hội tụ ổn định).

Phân tích

- **Công thức cập nhật**:

$$w \leftarrow w + \alpha(G_t - \hat{v}(s_t;w))\,\nabla_w \hat{v}(s_t;w).$$

  Khi dùng linear approximation $\hat{v}(s;w) = w^T\phi(s)$, thì $\nabla_w \hat{v}(s_t) = \phi(s_t)$ dễ tính.
- **Ưu điểm**: Unbiased, tức expectation của gradient estimate đúng bằng gradient của MSVE. Khi learning rate giảm dần phù hợp, hội tụ về local minimum của MSVE amreis.github.io datascience.stackexchange.com .
- **Nhược điểm**: Cần chờ đến khi episode kết thúc mới có G_t, không thích hợp khi episode dài hoặc vô hạn (infinite-horizon). Variance của G_t thường lớn, dẫn đến cập nhật nhiễu, làm chậm hội tụ và khó ổn định.

---

3. Temporal Difference Update cho Function Approximation

Slide trình bày "Temporal Difference Update":

- Thay vì target là return G_t, có thể dùng bootstrap target, ví dụ TD(0) target: $r_{t+1} + \gamma\hat{v}(s_{t+1};w)$. Ký hiệu chung target là U_t.

- **Nếu U_t là unbiased estimate**, trong điều kiện phù hợp, hàm xấp xỉ sẽ hội tụ. Trường hợp $U_t = G_t$ (Monte Carlo) là unbiased. Tuy nhiên, nếu U_t là bootstrap target dựa trên $\hat{v}(s_{t+1}; w)$, target phụ thuộc tham số w hiện tại, có thể gây bias.
- Slide nhấn: "TD update is not actually a stochastic gradient descent update":
  - Khi viết squared error cho một sample: $(U_t - \hat{v}(s_t; w))^2$, gradient đầy đủ với respect to w bao gồm cả đạo hàm của U_t nếu U_t phụ thuộc w. Nhưng TD bỏ qua phần này (chain rule bỏ phần derivative của target), do đó đây là **semi-gradient**.
  - Công thức cập nhật semi-gradient TD(0):

$$\delta_t = r_{t+1} + \gamma \hat{v}(s_{t+1}; w) - \hat{v}(s_t; w), \quad w \leftarrow w + \alpha\, \delta_t\, \nabla_w \hat{v}(s_t; w).$$

  - Vì target bao gồm $\hat{v}(s_{t+1}; w)$, nên cập nhật không phải là gradient descent trực tiếp trên MSVE, dẫn đến khả năng hội tụ đến điểm không phải local minimum của MSVE.
    learningtheory.org datascience.stackexchange.com .

Phân tích ưu/nhược điểm

- **Ưu điểm**:
  - Cập nhật mỗi bước thời gian, không cần chờ episode kết thúc → sample efficiency cao, có thể xử lý infinite-horizon hoặc môi trường online.
  - Variance thấp hơn Monte Carlo (bởi vì dùng ước lượng kế tiếp thay vì return dài)
    amreis.github.io web.eecs.umich.edu .
  - Thường hội tụ nhanh hơn Monte Carlo trong thực tế.
- **Nhược điểm**:
  - Có bias do bootstrap target không phải unbiased estimate. Với linear on-policy under certain conditions, vẫn hội tụ nhưng đến TD fixed point (không đảm bảo là local minimum MSVE) web.eecs.umich.edu learningtheory.org .
  - Với phi tuyến hoặc off-policy, dễ bất ổn, có thể divergence. Cần kỹ thuật bổ trợ (experience replay, target network, eligibility traces, hoặc các thuật toán ổn định như GTD, LSTD) để giảm thiểu vấn đề.
  - Learning rate phải chọn cẩn thận để cân bằng convergence và stability.

---

4. Bản chất "semi-gradient" của TD

- **Full gradient** cho MSVE: $\nabla_w [v_\pi(s) - \hat{v}(s; w)]^2$. Khi dùng MC, G_t là unbiased estimate của $v_\pi(s)$, và gradient descent dùng expectation của $(G_t - \hat{v}(s; w))\nabla_w \hat{v}(s; w)$ tương đương full gradient expectation.
- **Semi-gradient TD**: target $U_t = r_{t+1} + \gamma \hat{v}(s_{t+1}; w)$ chứa $\hat{v}(s_{t+1}; w)$. Nếu tính full gradient, phải thêm $\partial U_t / \partial w = \gamma \nabla_w \hat{v}(s_{t+1}; w)$. Nhưng TD ignore phần này, chỉ dùng "semi-gradient" $\nabla_w \hat{v}(s_t; w)$.

- Kết quả: cập nhật không hướng thẳng xuống local minimum MSVE, nhưng trong linear on-policy có lý thuyết chứng minh hội tụ đến TD fixed point (xem Sutton & Barto). Với phi tuyến, thiếu đảm bảo hội tụ chung, dễ gặp nhiều cạm bẫy. web.eecs.umich.edu datascience.stackexchange.com .

---

5. TD Converges to a Biased Estimate

- Vì semi-gradient, TD thường hội tụ đến TD fixed point, không nhất thiết minimal MSVE. Kết quả này tức là ước lượng có bias so với giá trị thật v_π. Trong linear on-policy, bias phụ thuộc cấu trúc feature và discount factor γ (khi γ gần 1, bias nhỏ hơn, nhưng convergence chậm hơn; khi γ nhỏ, bias lớn hơn nhưng convergence nhanh hơn). learningtheory.org web.eecs.umich.edu .
- Trong Monte Carlo, không có bootstrap, nên unbiased (trong expectation) với MSVE. TD trade-off bias – variance: chấp nhận một chút bias để giảm đáng kể variance và tăng tốc hội tụ.
- Nhiều nghiên cứu về bias-variance trade-off trong TD: ví dụ k-step TD hoặc λ-return, điều chỉnh tham số để cân bằng bias và variance tối ưu. learningtheory.org web.eecs.umich.edu .

---

6. So sánh Temporal Difference vs Monte Carlo

Slide liệt kê các khía cạnh:

1. **Update Timing**:
   - TD: cập nhật sau mỗi bước thời gian, dùng reward hiện tại và ước lượng next state.
   - MC: cập nhật sau khi hết cả episode, dùng full return.
2. **Bootstrapping**:
   - TD: bootstrap (dùng $\hat{v}(s_{t+1})$).
   - MC: không bootstrap, hoàn toàn dựa trên return thu được.
3. **Bias**:
   - TD: có bias (bootstrap target).
   - MC: unbiased (nhưng variance cao).
4. **Variance**:
   - TD: variance thấp hơn (cập nhật nhiều lần, chỉ dựa trên 1-step reward + giá trị ước lượng).
   - MC: variance cao hơn (return dài có thể dao động lớn).
5. **Sample Efficiency**:
   - TD: thường cần ít sample hơn để hội tụ gần đúng.
   - MC: cần nhiều sample, đặc biệt với episode dài hoặc reward thay đổi lớn.
6. **Ứng dụng**:
   - TD: thích hợp online, infinite-horizon, real-time, và function approximation.

- MC: thích hợp episodic tasks rõ ràng, khi có thể hoàn thành episode nhanh.

Phân tích

- **Thực tiễn**: Trong hầu hết các ứng dụng RL hiện đại (game, robotics, giao dịch…), TD (và các biến thể như TD(λ), Q-learning, SARSA, Actor-Critic) là phương pháp chủ đạo do sample efficiency và khả năng xử lý infinite-horizon. Monte Carlo thường dùng để khởi tạo benchmark hoặc trong trường hợp môi trường episodic nhỏ gọn. amreis.github.io web.eecs.umich.edu .
- **Bias-Variance Trade-off**: Khi environment ổn định, bias của TD có thể được chấp nhận đổi lấy giảm variance và tốc độ học nhanh. Trong các hệ thống phi tuyến (neural network), TD đôi khi dẫn đến instability, nhưng vẫn được dùng rộng rãi kèm kỹ thuật ổn định (replay buffer, target network, gradient clipping…).
- **Thiết kế thuật toán lai**: Nhiều phương pháp kết hợp MC và TD, ví dụ n-step returns, TD(λ) với eligibility traces, để điều chỉnh giữa hai cực này. Khi λ tăng đến 1, gần MC (giảm bias, tăng variance); λ nhỏ, gần TD(0) (tăng bias, giảm variance). Việc chọn λ phù hợp giúp cân bằng tùy theo môi trường. learningtheory.org web.eecs.umich.edu .

---

7. Tốc độ hội tụ của TD so với Gradient Monte Carlo

- Slide nhắc: “TD converges much faster than Gradient Monte Carlo”.
- **Giải thích**: Vì TD cập nhật mỗi bước và tận dụng bootstrap, nên giảm thiểu sai số ngay khi có reward cục bộ, đẩy thông tin về giá trị lan tỏa nhanh hơn trong chuỗi state. Monte Carlo phải thu thập toàn bộ return, nên độ trễ lớn hơn và variance cao, dẫn đến cập nhật chậm và ít ổn định hơn.
- **Nghiên cứu**: Các bài báo về bias-variance error bounds cho TD (Michael Kearns, Satinder Singh) chứng minh TD có upper-bound error nhanh giảm hơn Monte Carlo, và asymptotic error có thể chấp nhận. learningtheory.org web.eecs.umich.edu .
- **Ví dụ minh họa**: Trên Random Walk hoặc Gridworld, nếu update TD vs MC, thường thấy TD bình ổn tại giá trị ước lượng gần đúng nhanh hơn MC (nhất là với discount factor <1). Tuy ước lượng có bias, nhưng sau vài bước tương tác, mô hình đã có hình dung tương đối về value function, đủ để policy improvement.

---

8. Kết luận và khuyến nghị

- **Nắm rõ bản chất semi-gradient**: Hiểu rằng TD không tương đương gradient descent trên MSVE, dẫn đến bias. Tuy nhiên, trong linear on-policy có hội tụ ổn định đến TD fixed point.
- **Chọn phương pháp phù hợp**: Với không gian trạng thái nhỏ hoặc episodic ngắn, Monte Carlo vẫn có giá trị (đặc biệt khi cần unbiased estimate, ví dụ đánh giá chính sách cuối). Với môi trường online, infinite-horizon, hay khi cần sample efficiency, TD là lựa chọn chính.
- **Cân bằng bias – variance**: Sử dụng n-step returns hoặc λ-return để điều chỉnh giữa TD và MC, chọn λ dựa trên đặc tính môi trường và model complexity.

- **Function approximation phi tuyến**: Khi dùng neural network, TD vẫn được dùng rộng rãi (Deep Q-Network, Actor-Critic, v.v.), nhưng cần kỹ thuật ổn định: replay buffer, target networks, gradient clipping, normalization. Hiểu bias-variance và semi-gradient hữu ích để giải thích hiện tượng divergence hoặc oscillation.