1. Mối liên hệ giữa Tabular TD và Linear Semi-Gradient TD

- **Tabular TD**: Lưu giá trị ước lượng $V(s)$ riêng cho mỗi trạng thái s. Cập nhật TD(0) tabular:

$$V(s_t) \leftarrow V(s_t) + \alpha[r_{t+1} + \gamma V(s_{t+1}) - V(s_t)].$$

- **Linear Semi-Gradient TD**: Xấp xỉ giá trị bằng hàm tuyến tính trên feature:

$$\hat{v}(s; w) = w^\top \phi(s),$$

với vector feature $\phi(s)$. Cập nhật semi-gradient TD(0):

$$\delta_t = r_{t+1} + \gamma\, w^\top \phi(s_{t+1}) \;-\; w^\top \phi(s_t), \quad w \leftarrow w + \alpha\, \delta_t\, \phi(s_t).$$

- **Tabular là trường hợp đặc biệt**: nếu chọn feature one-hot cho mỗi trạng thái: $\phi_i(s) = 1$ nếu s là trạng thái thứ i, và 0 ngược lại, thì $\hat{v}(s; w) = w_i$. Khi cập nhật, $\phi(s_t)$ chọn đúng thành phần trọng số tương ứng w_i và công thức giống y hệt cập nhật tabular TD. Như vậy tabular TD = Linear TD với feature one-hot users.ece.cmu.edu .
- **Ý nghĩa**: Khung Linear TD cho phép mở rộng sang các feature tổng quát hơn (state aggregation, tile coding, RBF, embedding networks...), và Tabular TD là trường hợp đơn giản nhất.

---

2. Cập nhật Temporal Difference với Linear Function Approximation

- **Semi-gradient TD**: Khi target bootstrap phụ thuộc tham số w, ta dùng semi-gradient (bỏ phần đạo hàm của target). Cụ thể:

$$\delta_t = r_{t+1} + \gamma\, w^\top \phi(s_{t+1}) - w^\top \phi(s_t), \quad w \leftarrow w + \alpha\, \delta_t\, \phi(s_t).$$

- **Giải thích cập nhật**:
  - $\phi(s_t)$ là vector feature của trạng thái hiện tại. Nếu một thành phần feature lớn, cập nhật thay đổi tương ứng trọng số nhiều, ảnh hưởng lớn đến ước lượng. Nếu feature bằng 0, trọng số tương ứng không thay đổi.
  - Hướng cập nhật: nếu $\delta_t > 0$ (ước lượng hiện tại thấp hơn target bootstrap), ta tăng w theo hướng $\phi(s_t)$ để tăng $\hat{v}(s_t)$. Ngược lại nếu $\delta_t < 0$, giảm ước lượng.
- **Ưu/nhược**:
  - Ưu: cập nhật tại mỗi bước, không cần chờ hết episode. Dễ áp dụng online, infinite-horizon.
  - Nhược: bootstrap target tạo bias; cập nhật không phải gradient descent thuần túy trên MSVE. Với Linear On-policy và điều kiện A invertible, vẫn hội tụ đến một "TD fixed point" ocw.snu.ac.kr medium.com ; nhưng không tối ưu MSVE. Với phi tuyến, có thể không hội tụ ổn định.

---

3. Phân tích Expected TD Update, ma trận A và vector b

3.1. Kỳ vọng cập nhật (Expected update)

- Tại mỗi bước, weight cập nhật: $\Delta w_t = \alpha\, \delta_t\, \phi(s_t)$. Tuy nhiên do mẫu ngẫu nhiên, quan tâm đến **kỳ vọng** của Δw_t dưới phân phối trải nghiệm on-policy (steady-state). Khi hệ ổn định, ta giả sử phân phối trạng thái action ổn định, để viết:

$$\mathrm{E}[\Delta w_t] = \alpha(\mathrm{E}[r_{t+1}\phi(s_t)] + \gamma\, \mathrm{E}[\phi(s_t)\phi(s_{t+1})^\top]\, w - \mathrm{E}[\phi(s_t)\phi(s_t)^\top]\, w).$$

- **Đặt**

$$A = \mathrm{E}[\phi(s_t)(\phi(s_t) - \gamma\phi(s_{t+1}))^\top], \quad b = \mathrm{E}[r_{t+1}\phi(s_t)].$$

Khi đó kỳ vọng cập nhật:

$$\mathrm{E}[\Delta w_t] = \alpha(b - Aw).$$

Nếu bỏ noise (phần nhiễu xung quanh kỳ vọng), ta có view: cập nhật theo quy luật xấp xỉ gradient descent lên một objective liên quan đến A và b. ocw.snu.ac.kr users.ece.cmu.edu .

- **Giải thích A và b**:
  - A là ma trận d×d (d = chiều feature), thu từ các đặc trưng của trạng thái hiện tại và kế tiếp theo discount γ.
  - b liên quan đến reward và feature: trung bình reward nhân feature.

- **Điều kiện A invertible**: nếu A khả nghịch, có nghiệm duy nhất $w_{\text{TD}} = A^{-1}b$. Nếu không, tìm giải pháp tối ưu trong không gian con.

3.2. Điểm cố định (TD Fixed Point)

- **Định nghĩa**: Điểm cố định w_{TD} thỏa mãn kỳ vọng cập nhật bằng 0:

$$b - Aw_{\text{TD}} = 0 \quad \Longrightarrow \quad Aw_{\text{TD}} = b.$$

- **Ý nghĩa**: Khi w tới gần w_{TD}, kỳ vọng cập nhật xấp xỉ 0, tức trọng số ổn định theo trung bình. Trong Linear On-policy TD(0) với chọn step-size thích hợp, w hội tụ về w_{TD} (under standard conditions) ocw.snu.ac.kr .
- **Kết nối Bellman**: Hệ $Aw = b$ liên quan tới Bellman equation xấp xỉ. Thật ra, w_{TD} là nghiệm của projected Bellman equation trong không gian feature. Khung Linear TD lý giải tại sao ước lượng giá trị hội tụ về nghiệm của Bellman projector hơn là nghiệm chính xác trên toàn trạng thái (trong MSVE). medium.com .

---

4. Quan hệ giữa TD Fixed Point và Minimum Value Error Solution

- **Minimum Value Error (MSVE) solution**: Với mục tiêu MSVE:

$$J(w) = \mathrm{E}_{s\sim d}[v_\pi(s) - w^\top \phi(s)]^2,$$

  nghiệm tối ưu (linear least-squares) thỏa mãn $\mathrm{E}[\phi(s)\phi(s)^\top]w = \mathrm{E}[v_\pi(s)\phi(s)]$. Đặt ma trận $C = \mathrm{E}[\phi(s)\phi(s)^\top]$, vector $d = \mathrm{E}[v_\pi(s)\phi(s)]$, thì giải MSVE là $w_{\text{MSVE}} = C^{-1}d$ (nếu invertible).
- **So sánh với w_{TD}**:
  - Trong general, w_{TD} khác w_{MSVE}. Sự khác biệt phụ thuộc γ và cấu trúc feature. Khi γ gần 1, bias của TD fixed point so với MSVE có thể lớn; khi γ gần 0, TD fixed point gần nghiệm MSVE hơn. users.ece.cmu.edu ocw.snu.ac.kr .
  - **Ví dụ minh họa**:
    - Giả sử hai trạng thái đều được gộp chung feature (aggregation), true values khác nhau, như ví dụ Python ở dưới. Kết quả: w_{TD} chênh lệch đáng kể so với giá trị trung bình true (MSVE minimizer), đặc biệt nếu γ lớn. Khi γ nhỏ, ước lượng bootstrap phụ thuộc ít vào giá trị kế tiếp, do đó fixed point gần hơn average reward/cost cơ bản.
- **Ý nghĩa**: TD trade-off bias-variance: chấp nhận bias để có cập nhật bootstrap, giảm variance và tăng tốc học. Quan sát việc khác biệt fixed point vs MSVE giúp hiểu khi nào TD có thể sai lệch lớn.

---

5. Ví dụ minh họa Numeric

Để làm rõ, ta đưa ví dụ MDP đơn giản gồm 2 trạng thái non-terminal {0,1}:

- Từ state 0: chuyển sang state 1 với reward = 0.
- Từ state 1: chuyển về terminal (hoặc dừng) với reward = 1.
- Phân phối trạng thái $d(s)$ giả sử uniform {0,1}. Feature aggregation: $\phi(s) = 1$ cho cả hai trạng thái, tức chỉ một tham số w.
- Discount γ chọn giá trị (ví dụ 0.9).

  Ta tính:

$$A = \mathrm{E}[\phi(s)(\phi(s) - \gamma\phi(s'))] = 0.5 \times [1 \times (1 - \gamma \cdot 1)] + 0.5 \times [1 \times (1 - \gamma \cdot 0)] = 0.5(1-\gamma) + 0.5(1) = 0.5 - 0.5\gamma + 0.5 = 1 - 0.5\gamma.$$

  Với $\gamma = 0.9$, $A = 1 - 0.45 = 0.55$.

$$b = \mathrm{E}[r\phi(s)] = 0.5 \times 0 + 0.5 \times 1 = 0.5.$$

  Do đó TD fixed point $w_{\text{TD}} = b/A \approx 0.5/0.55 \approx 0.91$.
  Trong khi true values: $v(1) = 1$, $v(0) = \gamma \times 1 = 0.9$. MSVE minimizer (c sao cho minimize $(v(0) - c)^2 + (v(1) - c)^2$) là average $(0.9 + 1)/2 = 0.95$.
  Như vậy $w_{\text{TD}} \approx 0.91$ khác đáng kể so với 0.95, thể hiện bias do TD. Khi γ càng lớn (như 0.99), khác biệt càng lớn hơn, còn khi γ nhỏ, fixed point tiến gần average true hơn.
- **Mã minh họa** đã chạy qua python và hiển thị bảng giá trị, A, b, w_{TD} và MSVE minimizer để bạn quan sát trực quan.

---

6. Ảnh hưởng của feature và phân phối trạng thái

- **Chọn feature**: Nếu feature không đủ biểu diễn giá trị đúng (underfitting), TD fixed point sai lệch. Nếu feature quá cồng kềnh, có thể overfitting sample noise.
- **Phân phối trạng thái** $d(s)$: A và b đều phụ thuộc phân phối on-policy. Nếu policy thay đổi, A và b thay đổi ⇒ fixed point di chuyển. Khi training online, non-stationarity do policy improvement liên tục ảnh hưởng đến stabilité of w.
- **State aggregation**: Là trường hợp đặc biệt linear feature one-hot nhóm: gộp nhiều trạng thái chung feature. Cập nhật TD cho một trạng thái trong nhóm sẽ đồng thời ảnh hưởng đến tất cả trạng thái nhóm, tạo generalization nhưng dễ bias nếu nhóm không đồng nhất.

---

7. Thuật toán và hội tụ

- **Hội tụ Linear On-policy TD(0)**: Khi A ma trận d×d là positive definite (điều kiện tùy feature và phân phối trải nghiệm), với step-size α_t đủ nhỏ và giảm dần theo điều kiện SA (stochastic approximation), w hội tụ về $w_{\text{TD}} = A^{-1}b$ gần chắc chắn. ocw.snu.ac.kr medium.com .
- **Ý nghĩa positive definite**: Thường đòi hỏi feature vector bounded và phân phối đủ khám phá trạng thái, tránh A bị số một chiều hay không invertible.
- **Objective ẩn**: Mặc dù cập nhật không đầy đủ gradient MSVE, vẫn có thể định nghĩa một objective (Bellman error projected) mà linear TD xấp xỉ tối ưu. Tham khảo Least-Squares Temporal Difference (LSTD) để tìm nghiệm trực tiếp mà bỏ qua cập nhật gradient ngẫu nhiên.

---

8. Ảnh hưởng của giá trị gamma

- **Gamma gần 1**: TD fixed point khác xa MSVE minimizer; bias lớn, nhưng vì bootstrap lồng sâu, cập nhật nhạy vào giá trị kế tiếp, sample efficiency cao, nhưng nếu phi tuyến, dễ instable.
- **Gamma nhỏ**: Fixed point gần MSVE; bias nhỏ; bootstrap ít phụ thuộc giá trị kế tiếp; cập nhật tập trung vào reward ngắn hạn; nhưng cắt giảm tầm nhìn dài hạn, có thể không học được giá trị tương lai quan trọng.
- **Lựa chọn gamma**: Tùy bài toán, cân bằng tầm nhìn dài-ngắn hạn; trong thực tế thường chọn gamma gần 1 để quan tâm dài hạn, nhưng chấp nhận bias TD và áp dụng kỹ thuật ổn định (target network, replay buffer trong Deep RL).

---

9. Kết hợp và ứng dụng thực tế

- **Tabular vs Linear**: Tabular phù hợp khi không gian trạng thái nhỏ. Khi không gian lớn/continuous, phải dùng linear hoặc phi tuyến.
- **State aggregation & feature engineering**: Thiết kế feature quyết định chất lượng xấp xỉ giá trị. Ví dụ tile coding, coarse coding giúp linear xấp xỉ phi tuyến hiệu quả.
- **LSTD, Gradient-TD**: Khi cần tìm w_{TD} chính xác, có thể dùng LSTD trực tiếp giải $Aw = b$ (yêu cầu lưu/truy cập batch dữ liệu hoặc ước lượng A,b). Với streaming data, TD update dạng stochastic vẫn được dùng.
- **Deep RL**: Mặc dù không thuần linear, nhưng khái niệm TD fixed point và semi-gradient vẫn quan trọng để hiểu nguyên nhân instability khi dùng neural network. Kỹ thuật như target network, double Q-learning, prioritized replay… nhằm giảm chệch và ổn định cập nhật bootstrap.
- **Policy evaluation và Actor-Critic**: Critic thường dùng Linear/Deep TD để ước lượng value; actor dựa vào critic để cập nhật chính sách. Hiểu fixed point giúp debug khi giá trị ước lượng sai lệch.

---

10. Tóm tắt

- **Tabular TD là trường hợp đặc biệt** của Linear TD với feature one-hot; Linear TD mở rộng cho feature tổng quát.
- **Semi-gradient TD update**: cập nhật theo $\delta_t \phi(s_t)$, thuận tiện online nhưng tạo bias.
- **Expected TD update**: phân tích thành $b - Aw$.
- **TD Fixed Point**: nghiệm $Aw = b$. Khi A khả nghịch, $w_{\text{TD}} = A^{-1}b$.
- **So sánh với MSVE minimizer**: khác nhau, chênh lệch tăng khi γ gần 1.
- **Hội tụ**: Linear On-policy TD(0) hội tụ về điểm cố định nếu điều kiện phù hợp.

- **Ứng dụng**: Cơ sở để xây dựng algorithms RL phức tạp hơn (Deep RL, Actor-Critic...), cũng như hiểu trade-off bias-variance.