1. **Hiểu average reward**: Khác với discounted return, average reward tập trung tối ưu tốc độ reward trung bình dài hạn, đặc biệt trong các bài toán tiếp diễn (continuing tasks) không có ranh giới tập kết thúc rõ ràng.
2. **Hiểu differential value functions**: Khi dùng average reward, tiêu chuẩn đánh giá giá trị của trạng thái cần điều chỉnh (differential) vì không dùng γ<1 để làm hội tụ. Differential value function và differential action-value function là những khái niệm then chốt để xây dựng Bellman equation và thuật toán học. lcalem.github.io ocw.snu.ac.kr .

---

2. Khái niệm Average Reward và sự khác biệt so với Discounted Reward

2.1. Discounted Return vs Average Reward

- Trong hầu hết RL truyền thống, **mục tiêu** là tối đa hóa expected discounted cumulative reward:

$$G_t = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}, \quad 0 \le \gamma < 1.$$

  Discounting giúp đảm bảo hội tụ (return hữu hạn) và nhấn mạnh reward gần thời điểm hiện tại hơn reward xa. Tuy nhiên với các tác vụ tiếp diễn vô hạn hoặc không xác định độ dài episode, việc chọn γ phù hợp đôi khi khó: γ quá thấp khiến agent myopic, bỏ qua reward dài hạn; γ quá gần 1 dẫn đến variance cao, khó học và có thể return tiệm cận vô hạn. bedirtapkan.com .

- **Average reward formulation** đặt mục tiêu tối đa hóa reward trung bình trên mỗi bước thời gian trong vô hạn:

$$\rho(\pi) = \lim_{T\to\infty} \frac{1}{T} \mathrm{E}[\sum_{t=1}^{T} R_t],$$

  hoặc trong phân phối trạng thái dần đến ổn định under policy π:

$$\rho(\pi) = \sum_s \mu_\pi(s) \sum_a \pi(a|s)\, r(s,a),$$

  với μπ(s) là tần suất xuất hiện trạng thái s theo policy π trong dài hạn lcalem.github.io bedirtapkan.com .

- **Ưu điểm average reward**:
  - Phù hợp với continuing tasks không có điểm kết thúc rõ ràng (ví dụ điều khiển liên tục, scheduling).
  - Không cần chọn γ; agent quan tâm đồng đều reward gần và xa, hơn là cắt discount.
  - Đơn giản trong mục tiêu: tối thiểu hóa hoặc tối đa hóa rate per step.
- **Nhược điểm**:
  - Không phân biệt ưu tiên reward gần hay xa: trong một số bài toán, hành vi myopic (ưu tiên reward trước mắt) lại phù hợp, average reward có thể làm agent bỏ qua reward ngắn

hạn quan trọng.

- Việc học average reward và differential value function thường phức tạp hơn về mặt thuật toán, có thể dẫn đến khó hội tụ hoặc chậm học hơn so với discounted TD trong một số môi trường.
- Khi môi trường non-stationary, average reward formulation cần cập nhật liên tục, có thể phức tạp. bedirtapkan.com .

---

3. Ví dụ Ring trái / Ring phải và phân tích γ

Slide nêu ví dụ: MDP đa phần ở mỗi state chỉ có một hành động duy nhất, ngoại trừ một state S tại đó agent có thể chọn "traversing left ring" hoặc "traversing right ring". Cấu trúc reward:

- Trên left ring: có một reward +1 ngay sau state S.
- Trên right ring: có reward +2 ngay trước khi quay trở lại state S.
  Tức nếu đi vòng trái, nhận +1 tại bước đầu; nếu đi vòng phải, nhận +2 nhưng muộn hơn, tức reward xa hơn trong chuỗi.

3.1. Đánh giá theo discounted return

- Nếu dùng discounted return, agent so sánh:
  - Left: nhận +1 ngay → giá trị ≈ 1.
  - Right: nhận +2 sau một số bước (ví dụ độ dài vòng m bước) → giá trị ≈ γ^m * 2.
- Agent chọn left nếu $1 > \gamma^m \cdot 2$, tức γ^m < 0.5 ⇒ γ < (0.5)^(1/m). Với m lớn, hoặc γ nhỏ, left được ưu tiên do nhận reward sớm, discounted return cao hơn.
- Nếu γ đủ lớn, có thể γ^m *2 >1, agent chọn right do tổng discounted return lớn hơn, dù reward đến muộn. Slide nêu: "we can figure out the minimum value of gamma so that the agent prefers the policy that goes right. Gamma needs to be at least 0.841." Con số này phụ thuộc m; trong ví dụ slide, có thể m=1 hoặc 2 dẫn đến kết quả 0.841. Đây là ví dụ minh hoạ: với discount, agent có thể myopic nếu γ không đủ cao. Khi γ thấp, left ưa thích; khi γ cao, right ưa thích. datacamp.com .
- Khi γ → 1, discounted return gần giống sum undiscounted, agent quan tâm reward dài hạn, chọn right (reward lớn hơn tổng theo chu kỳ). Tuy nhiên không thể γ=1 trong continuing setting nếu tổng reward vô hạn hoặc không hội tụ.

3.2. Khó khăn khi chọn γ

- Nếu γ quá nhỏ, agent quá myopic, bỏ qua reward có giá trị cao nhưng đến sau.
- Nếu γ quá gần 1, return dài và variance lớn, learning khó ổn định.
- Dùng average reward sẽ tránh phải tinh chỉnh γ, vì agent trực tiếp tối ưu rate reward trung bình: agent sẽ chọn vòng có reward trung bình mỗi bước cao nhất, ví dụ giữa left ring (giả sử reward 1 mỗi m bước → rate = 1/m) và right ring (reward 2 mỗi m bước → rate = 2/m), rõ ràng right ring tốt. Average reward khẳng định chọn right mà không cần γ. lcalem.github.io .

4. Công thức Average Reward qua phân phối trạng thái

4.1. Định nghĩa average reward ρ

Trong mô hình MDP tiếp diễn, giả sử policy π định hướng hành động. Khi chạy trong thời gian dài, trạng thái xuất hiện theo phân phối ổn định μπ(s). Tại mỗi bước, reward kỳ vọng khi ở state s và chọn action a là r(s,a). Do đó average reward:

$$\rho(\pi) = \sum_s \mu_\pi(s) \sum_a \pi(a|s)\, r(s,a).$$

Nếu biết μπ(s), ta tính trực tiếp. Trong thực tế, μπ(s) ước lượng qua tương tác dài. lcalem.github.io .

4.2. Ý nghĩa

- ρ(π) là expected reward trung bình trên mỗi bước khi agent chạy dài lâu theo π. Agent muốn tìm π tối đa ρ.
- Không dùng discount, nên không ưu tiên reward gần hay xa. Thích hợp khi quan tâm hiệu suất dài hạn ổn định.

---

5. Differential Value Functions và Bellman Equation

Khi bỏ discount, ta không thể sử dụng giá trị thông thường $V(s) = \E[\sum \gamma^k R_{t+k+1}]$ vì γ=1 dẫn đến return vô hạn. Thay vào đó dùng **differential return** (hoặc differential value):

$$G_t^{\text{diff}} = \sum_{k=0}^{\infty} (R_{t+k+1} - \rho(\pi)).$$

Điều này nhằm đo lường phần "thặng dư" reward so với average reward mỗi bước. Differential state-value function:

$$h_\pi(s) = \E_\pi[\sum_{k=0}^{\infty} (R_{t+k+1} - \rho(\pi)) \mid S_t = s].$$

Tương tự action-value:

$$q_\pi(s,a) = \E_\pi[\sum_{k=0}^{\infty} (R_{t+k+1} - \rho(\pi)) \mid S_t = s, A_t = a].$$

Bellman equation cho differential giá trị:

$$h_\pi(s) = \sum_a \pi(a|s)[r(s,a) - \rho(\pi) + \sum_{s'} p(s'|s,a)\, h_\pi(s')].$$

Tương tự với q:

$$q_\pi(s,a) = r(s,a) - \rho(\pi) + \sum_{s'} p(s'|s,a) \sum_{a'} \pi(a'|s')\, q_\pi(s',a').$$

Đây là Bellman equation trong average reward setting web.stanford.edu .

- Chú ý: hπ(s) không duy nhất, có thể cộng một hằng; thường chọn một reference state hoặc constraint để xác định giá trị duy nhất (ví dụ fix hπ(s_ref)=0). Nhưng quan trọng là differences giữa states.
- Differential advantage: Aπ(s,a) = qπ(s,a) − Vπ(s) (với Vπ(s)=hπ(s) variant).

---

6. Thuật toán Differential Semi-gradient Sarsa (Average Reward Sarsa)

Dùng để ước lượng qπ và ρ khi chạy on-policy trong continuing tasks với function approximation. Cơ bản giống Sarsa nhưng:

- Thay target reward bằng (R_{t+1} − ρ_estimate).
- Đồng thời cần cập nhật ước lượng average reward ρ.

6.1. Công thức thuần túy (tabular)

Pseudo-code cho tabular differential Sarsa:

1. Khởi Q(s,a) có thể zero, khởi estimate ρ = 0.
2. Chạy theo policy π (ví dụ ε-greedy dựa trên Q). Với mỗi bước t:
   - Quan sát (S_t, A_t, R_{t+1}, S_{t+1}, A_{t+1}).
   - Tính TD error differential:

$$\delta_t = R_{t+1} - \rho + Q(S_{t+1}, A_{t+1}) - Q(S_t, A_t).$$

   - Cập nhật Q:

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha\, \delta_t.$$

   - Cập nhật ước lượng ρ:

$$\rho \leftarrow \rho + \beta\, \delta_t,$$

     với β là step-size riêng cho average reward.
   - Nếu S_{t+1} terminal? Trong continuing tasks không có terminal, thường xử lý liên tục.
3. Lặp vô hạn.
   Thuật toán đảm bảo Q và ρ hội tụ (dưới điều kiện ergodic MDP, đủ explore, step-sizes phù hợp) đến differential qπ và average reward ρ(π). datascience.stackexchange.com .

6.2. Semi-gradient với function approximation

Khi xấp xỉ Q(s,a; w) = wᵀ φ(s,a) (linear) hoặc network, cập nhật semi-gradient:

- Tại bước t, có estimate $\hat{q}(S_t, A_t; w)$ và average reward estimate $\bar{R}$.
- Chọn A_{t+1} theo policy on-policy (e.g., ε-greedy trên $\hat{q}$).
- Tính TD error differential:

$$\delta_t = R_{t+1} - \bar{R} + \hat{q}(S_{t+1}, A_{t+1}; w) - \hat{q}(S_t, A_t; w).$$

- Cập nhật weights:

$$w \leftarrow w + \alpha\, \delta_t\, \nabla_w \hat{q}(S_t, A_t; w).$$

  Với linear: $\nabla_w \hat{q}(S_t, A_t) = \phi(S_t, A_t)$.

- Cập nhật average reward:

$$\bar{R} \leftarrow \bar{R} + \beta\, \delta_t.$$

  β thường nhỏ hơn α hoặc điều chỉnh theo lý thuyết stochastic approximation (β step-size on slower timescale so that w có thể theo kịp). datascience.stackexchange.com .

- Nếu dùng neural network, gradient là backprop; update ρ tương tự một tham số scalar.
- **Giải thích**: Thay vì bootstrap target là R + γ Q(next), ta dùng R - ρ + Q(next), tương đương γ=1 và trừ ρ để đảm bảo convergence. ρ đóng vai baseline: nếu R_{t+1} > ρ, δ>0, Q(S_t,A_t) tăng; nếu R_{t+1}<ρ, δ<0, Q giảm. Qua thời gian dài, ρ ước lượng average reward, giúp differential returns hội tụ. datascience.stackexchange.com .

6.3. Convergence và điều kiện

- Yêu cầu MDP ergodic: policy đảm bảo visit mọi state-action đủ thường xuyên, để average reward và differential q well-defined.
- Step-sizes: α and β phải giảm dần theo điều kiện stochastic approximation ($\sum\alpha=\infty$, $\sum\alpha^2<\infty$, tương tự β), hoặc dùng small constant với lý thuyết convergence gián đoạn. Thường β < α (cập nhật ρ chậm hơn hoặc cùng tốc độ tuỳ lý thuyết).
- Với linear approximation, under on-policy and đủ điều kiện, thuật toán hội tụ đến nghiệm của projected Bellman equation cho differential q. Với phi tuyến (NN), tương tự khó ổn định hơn nhưng thường áp dụng trong deep RL with tricks.
- **Tham khảo lý thuyết**: Sutton & Barto mục "Average rewards: a new problem setting for continuing tasks" và "Differential Semi-gradient Sarsa" trong Chương 10 lcalem.github.io ocw.snu.ac.kr .

---

7. Ví dụ minh họa trong slide

Slide nêu ví dụ ring trái/phải để giải thích discounted setting có giới hạn vì phải chọn γ lớn đến gần 1 để ưu tiên reward dài, trong khi average reward trực tiếp so sánh rate reward qua vòng. Ngoài ra slide có thể dẫn đến differential Sarsa áp dụng trong ví dụ queuing hoặc bài toán tiếp diễn đi theo vòng lặp.

Ví dụ minh hoạ Differential Sarsa (tiếp diễn):

- Môi trường: agent liên tục thực hiện các bước, tại mỗi vòng quay có reward khác nhau.
- Bắt đầu khởi Q và ρ. Agent thử left hoặc right. Khi theo policy on-policy (ví dụ ε-greedy), update differential TD error mỗi bước: R - ρ + Q(next) - Q(current). Qua thời gian, ρ ước lượng rate reward dài hạn. Agent sẽ học ưu tiên action đem lại differential positive lâu dài (ví dụ ring phải nếu reward rate cao hơn).

---

8. Ưu và nhược điểm Average Reward & Differential Methods

8.1. Ưu điểm

- **Thích hợp continuing tasks**: Không cần phân chia episode rõ ràng hay chọn γ; agent tập trung tối ưu rate dài hạn.
- **Loại bỏ vấn đề chọn γ**: Tránh phải điều chỉnh γ rất gần 1, giảm variance và khó học trong discounted case.
- **Bellman equation differential**: Có công thức tương tự discounted nhưng thay γ=1 và thêm baseline ρ, giúp thiết lập update TD tương tự Sarsa, Q-learning nhưng trong setting vô hạn.
- **Ứng dụng**: Trong các bài toán scheduling, điều khiển liên tục, điều phối tài nguyên, nơi không có ranh giới kết thúc tự nhiên.

8.2. Nhược điểm và thách thức

- **Complexity**: Cần ước lượng thêm average reward scalar ρ, và differential value functions có thể không trực quan giống discounted.
- **Convergence khó khăn hơn**: Trong một số môi trường hoặc khi function approximation phi tuyến, dễ gặp instability. Cần đảm bảo ergodicity, exploration thỏa đáng, step-size scheduling hợp lý.
- **Không ưu tiên reward ngắn hạn**: Khi immediate reward quan trọng hơn hoặc cần trade-off giữa gần và xa, average reward có thể không biểu diễn đúng objective mong muốn.
- **Khó xác định reference**: Differential value function định nghĩa chỉ tới hằng additive, cần chọn reference state hoặc constraint để xác định unique solution; tuy nhiên trong learning thường không cần gán cố định, chỉ quan tâm update difference.
- **Phổ biến thực nghiệm**: Discounted methods (với γ<1) vẫn phổ biến hơn, nhiều thư viện hỗ trợ; average reward và differential algorithms ít được dùng rộng rãi, đòi hỏi hiểu sâu hơn và custom implementation. lcalem.github.io datascience.stackexchange.com .

---

9. Gợi ý triển khai và thử nghiệm

1. **Xác định môi trường suitable**: Nếu task continuing, không rõ độ dài episode, reward rate mới quan trọng, cân nhắc average reward. Nếu episodic rõ ràng hoặc reward gần quan trọng, vẫn dùng discounted.

2. **Xây dựng differential Sarsa**:
    - Với linear approximation: lựa chọn feature φ(s,a) (stacked nếu discrete actions) hoặc network nếu cần.
    - Khởi ρ và w; chọn step-sizes α, β.
    - Chạy on-policy (ε-greedy) liên tục.
    - Theo dõi ρ estimate qua thời gian, theo dõi differential Q để đảm bảo hội tụ.
3. **Thử so sánh với discounted Sarsa**: Chạy trên cùng môi trường với γ rất gần 1, so sánh performance và stability learning.
4. **Xác minh ergodicity / exploration**: đảm bảo policy khám phá đủ để hỗ trợ average reward ước lượng chính xác.
5. **Tuning step-sizes**: α và β phải cân bằng; thường β nhỏ hơn α hoặc cùng tốc độ chậm.
6. **Monitoring**: Theo dõi ρ(t) convergence; theo dõi return trung bình per step trong thực thi policy learned để kiểm tra improvement.
7. **Debug**: Kiểm tra δ_t tính đúng R - ρ + Q(next) - Q(current); kiểm tra cập nhật ρ; xem ρ có drift ổn định không; feature representation có thể ảnh hưởng ổn định.
8. **Mở rộng**: Có thể triển khai n-step differential Sarsa, differential TD(λ), differential Q-learning (off-policy) với average reward, nhưng off-policy average reward hay differential Q-learning phức tạp hơn và ít lý thuyết mạnh.

---

10. Tóm tắt

- **Average reward** là objective tối ưu rate reward dài hạn, phù hợp continuing tasks, tránh phải chọn γ gần 1.
- **Differential value function** định nghĩa return là tổng (R - ρ) vô hạn, dẫn đến Bellman equation differential.
- **Differential Semi-gradient Sarsa**: thuật toán on-policy learning Q và ρ song hành, update theo δ = R - ρ + Q(next) - Q(current).
- **Ưu/nhược**: average reward thích hợp với continuing, nhưng cần ước lượng thêm ρ, có thể học phức tạp hơn; không ưu tiên reward ngay; implementation và convergence cần chú ý exploration và step-sizes.
- **Ví dụ ring** minh họa hạn chế của discounted nếu γ không đủ lớn, còn average reward trực tiếp chọn policy tối ưu rate reward.
- Khi triển khai, so sánh với discounted case để quyết định setting phù hợp.