Using Monte Carlo for Action Values

- **Học giá trị hành động** bằng Monte Carlo tương tự như học giá trị trạng thái:
    - Thay vì chỉ tính giá trị của trạng thái, ta tính **giá trị của cặp trạng thái–hành động (state-action)** bằng cách **lấy trung bình các return thu được khi theo chính sách từ cặp đó**.
- **Tại sao cần học giá trị hành động?**
    - Giúp **so sánh các hành động trong cùng một trạng thái**.
    - Từ đó có thể **chuyển sang hành động tốt hơn**, nếu có.
- **Vấn đề về khám phá (exploration):**
    - Nếu một hành động **không bao giờ được chọn bởi chính sách hiện tại**, ta **không thể ước lượng giá trị của nó**.
    - Điều này khiến ta **không học được đầy đủ giá trị của tất cả hành động** → gây sai lệch trong việc đánh giá và cải thiện chính sách.
- **Giải pháp 1: Exploring Starts (Khởi đầu khám phá):**
    - Bắt đầu mỗi tập bằng **một cặp trạng thái–hành động được chọn ngẫu nhiên**.
    - Sau hành động đầu tiên, agent sẽ **làm theo chính sách hiện tại**.
    - Yêu cầu: có thể **kiểm soát điểm bắt đầu** của mỗi tập → phù hợp trong môi trường như **grid world**.
- **Giải pháp 2 (gợi ý):**
    - **Epsilon-greedy** là chiến lược khám phá khác, **phù hợp với chính sách ngẫu nhiên (stochastic policies)**. Sẽ được trình bày kỹ hơn sau.
- **Kết luận:**
    - Monte Carlo có thể dùng để **ước lượng giá trị hành động**.
    - Tuy nhiên, cần **duy trì sự khám phá** để đảm bảo agent học đầy đủ và chính xác.

Using Monte Carlo methods for generalized policy iteration

- **GPI – Generalized Policy Iteration** là quá trình gồm hai bước lặp lại:
    - **Đánh giá chính sách (Policy Evaluation):** Ước lượng giá trị hành động (action-value) của chính sách hiện tại.
    - **Cải thiện chính sách (Policy Improvement):** Cập nhật chính sách bằng cách **chọn hành động greedy** với các giá trị hành động đã học.
- **Monte Carlo trong GPI:**
    - Sử dụng Monte Carlo để thực hiện **policy evaluation → ước lượng giá trị hành động Q(s, a)** bằng cách lấy **trung bình các returns** sau nhiều lần trải nghiệm từ mỗi cặp (trạng thái, hành động).
    - Sau mỗi tập (episode), cập nhật chính sách bằng cách **chọn hành động có Q cao nhất** trong mỗi trạng thái đã thấy → thực hiện **policy improvement**.
- **Exploring Starts (Khởi đầu khám phá):**
    - Mỗi tập được bắt đầu từ **một cặp trạng thái–hành động ngẫu nhiên** → đảm bảo mọi hành động đều được khám phá và đánh giá.
    - Điều kiện cần để **Monte Carlo đảm bảo hội tụ**.
- **Yêu cầu của GPI:**
    - Không cần ước lượng chính xác toàn bộ Q(s, a), chỉ cần các ước lượng **cải thiện dần theo thời gian** là đủ để đảm bảo hội tụ.
- **Monte Carlo Control with Exploring Starts:**
    - Một ví dụ cụ thể về GPI sử dụng Monte Carlo.
    - Kết hợp cả hai bước **policy evaluation** và **policy improvement sau mỗi episode**.

Epsilon-soft policies

1. Vấn đề với Exploring Starts:

- **Yêu cầu khắt khe:** Thuật toán cần bắt đầu từ **mọi cặp trạng thái–hành động**, điều này **khó hoặc không khả thi** trong nhiều bài toán thực tế.

- **Ví dụ:** Để dạy một xe tự lái học từ mọi trạng thái–hành động có thể, ta sẽ phải **đặt xe vào hàng ngàn tình huống nguy hiểm** → bất khả thi và không an toàn.

---

2. Giải pháp: Epsilon-Greedy (Epsilon-Soft)

- **Epsilon-Greedy:** Chính sách chọn hành động tốt nhất (greedy) **đa số thời gian**, nhưng **thi thoảng chọn hành động ngẫu nhiên**.

- **Epsilon-Soft:** Chính sách mà **mỗi hành động đều có xác suất ≥ ε / số hành động**. Đây là **tập con của các chính sách ngẫu nhiên (stochastic policies)**.

---

3. Lợi ích của Epsilon-Soft:

- **Duy trì khám phá liên tục:** Vì luôn có xác suất để thử mọi hành động, agent **sẽ dần trải nghiệm hết mọi trạng thái–hành động**.

- **Không cần Exploring Starts nữa.**

- Tuy không tìm được **chính sách tối ưu hoàn toàn**, nhưng tìm được **chính sách tối ưu trong tập Epsilon-Soft**.

---

4. So sánh với Chính sách Deterministic:

- Chính sách **deterministic** luôn chọn cùng một hành động → **thiếu khám phá**.

- Chính sách **epsilon-greedy** tạo ra hành vi đa dạng hơn → **khám phá tốt hơn và học được nhiều hơn**.

---

5. Thuật toán Monte Carlo Control với Epsilon-Soft:

- **Khởi tạo:** Chính sách ban đầu là epsilon-soft (ví dụ: chính sách ngẫu nhiên).

- **Sinh dữ liệu:** Agent **theo chính sách epsilon-soft** để tạo tập.

- **Đánh giá chính sách:** Cập nhật giá trị Q bằng trung bình return.

- **Cải thiện chính sách:** Cập nhật chính sách thành **epsilon-greedy theo Q hiện tại**.

✅6. Kết luận:

- Dù không đạt chính sách tối ưu tuyệt đối, **Monte Carlo với epsilon-soft** giúp **học được chính sách tốt gần tối ưu**, đồng thời **loại bỏ rào cản từ Exploring Starts**.

- Trong tương lai, các phương pháp như **Q-learning** sẽ giúp tìm chính sách tối ưu thực sự.