

2. Khái niệm Exploration và Exploitation


- **Exploration (Khám phá):**
 - Là quá trình thăm dò các hành động mới hoặc ít thử trước đó, nhằm **thu thập thêm thông tin** về môi trường.
 - Mục đích: phát hiện những hành động có thể cho reward cao hơn in the long run, ngay cả khi hiện tại chưa chắc chắn.
 - Kết quả: có thể tạm chấp nhận reward thấp lúc đầu để đổi lấy kiến thức về hành động mới, từ đó tăng tổng reward trong tương lai. 📄
 - **Exploitation (Khai thác):**
 - Là việc **tận dụng** kiến thức hiện có (các hành động đã được ước lượng có reward kỳ vọng cao) để tối đa hóa reward ngay lập tức.
 - Mục đích: tận dụng "hành động tốt nhất đã biết" để thu được reward cao ngay tại bước hiện tại.
 - Hạn chế: nếu chỉ khai thác, agent có thể bỏ sót những hành động tiềm năng khác mà chưa từng thử, dẫn đến khả năng bỏ qua lựa chọn tốt hơn. 📄
-

3. Ví dụ minh họa cơ bản


- **Tình huống thực tế (Mr. Mạnh và bữa ăn với thầy Hoa):**
 1. Mỗi khi đến nhà hàng, Mr. Mạnh đều gọi cùng một món (hành động khai thác), vì đã biết món đó ngon.
 2. Tuy nhiên, nhà hàng có nhiều món mới (hành động khám phá) trông rất hấp dẫn nhưng chưa có thông tin chắc chắn.
 3. Vấn đề: nếu luôn gọi món cũ, ông sẽ không biết món mới có ngon hơn hay không; nếu chỉ thử món mới, có thể ăn không vừa miệng và mất reward an toàn.
 4. Do đó, ông cần cân bằng: phần lớn thời gian gọi món cũ để chắc chắn có bữa ăn ngon (exploitation), nhưng thỉnh thoảng thử món mới để xem liệu có món nào còn ngon hơn (exploration). 📄
-

4. Cách biểu diễn hành vi Exploration


- Giả sử trong nhà hàng có nhiều "đĩa ăn" (mỗi đĩa tương ứng với một hành động).
- **Biến số chính:**
 - $Q(a)$: ước lượng giá trị (estimated value) của đĩa ăn a .
 - $N(a)$: số lần đã chọn đĩa ăn a .

- $q_*(a)$: giá trị thực sự (true value) của đĩa ăn a (thông thường ẩn, không biết trước).
 - **Chiến lược Round Robin (ví dụ khám phá thuần túy):**
 1. Mỗi lần đến, Mr. Mạnh chọn một đĩa khác nhau theo thứ tự vòng tròn, đảm bảo rằng mỗi món đều được thử ít nhất một lần.
 2. Sau một khoảng thời gian, ông sẽ dần biết giá trị $Q(a)$ cho mỗi món và xác định được món ngon nhất.
 3. Ví dụ:
 - Lần 1: chọn món A, nhận reward +1 \Rightarrow cập nhật $Q(A)$.
 - Lần 2: chọn B, nhận reward +3 \Rightarrow cập nhật $Q(B)$.
 - Lần 3: chọn C, nhận reward -1 \Rightarrow cập nhật $Q(C)$.
 - Lần 4: chọn A, nhận reward +3 \Rightarrow cập nhật thêm cho $Q(A)$.
 - ...
 4. Sau vài lần, các $Q(a)$ đủ ổn định, ông sẽ biết rõ món ngon nhất (giá trị thực q_* cao nhất).
-

5. Cách biểu diễn hành vi Exploitation

- Khi $Q(a)$ ước lượng đã phản ánh gần đúng các giá trị thực, ông sẽ chỉ **luôn chọn món có $Q(a)$ lớn nhất** (greedy).
 - Ví dụ:
 1. Giả sử sau một thời gian, ông đo được:
 - $Q(A) = +5$; $Q(B) = +3$; $Q(C) = 0$.
 2. Lần tới, ông sẽ chỉ gọi món A (món có Q cao nhất), bất chấp vẫn chưa biết rõ món D, E,... có ngon hơn không.
 3. Hậu quả: nếu có món F ngon hơn A nhưng chưa từng thử, ông sẽ mãi không phát hiện ra, vì luôn khai thác vào A. 
-

6. Vấn đề chính: Không thể vừa khám phá vừa khai thác đồng thời

- **Mâu thuẫn:**
 - **Exploration:** ưu tiên thu thập thông tin (có thể gây mất reward ngay tức thì).
 - **Exploitation:** ưu tiên thu reward ngay bây giờ (có thể bỏ lỡ thông tin quan trọng).
 - Do đó, cần một **chính sách** để **xác định thời điểm** và **tần suất** chuyển từ khám phá sang khai thác sao cho tối ưu tổng reward lâu dài. 

7. Phương pháp Epsilon-Greedy

- **Ý tưởng chính:** Với xác suất nhỏ ε , thực hiện một hành động ngẫu nhiên (khám phá); với xác suất còn lại $(1 - \varepsilon)$, chọn hành động có ước lượng $Q(a)$ lớn nhất (khai thác).
- **Công thức** (theo thời điểm t):

$$A_t = \begin{cases} \text{random action} & \text{với xác suất } \varepsilon, \\ \arg \max_a Q_t(a) & \text{với xác suất } (1 - \varepsilon). \end{cases}$$

- Trong đó A_t là hành động được chọn tại bước t .
- $Q_t(a)$ là ước lượng giá trị hành động tại bước t .
- **Giải thích chi tiết:**
 1. **Khai thác** (probability $1 - \varepsilon$) – chọn hành động tốt nhất theo ước lượng hiện tại, nhằm tối đa reward ngay.
 2. **Khám phá** (probability ε) – chọn ngẫu nhiên bất kỳ action nào, kể cả action có ước lượng thấp, mục đích thu thêm thông tin.
 3. Thông thường ε được đặt rất nhỏ (ví dụ 0.1 hoặc 0.01) để chủ yếu khai thác, nhưng vẫn đảm bảo khám phá đủ. 📄
- **Minh họa bằng con xúc xắc:**
 - Nếu muốn $\varepsilon = \frac{1}{6}$, ta gieo con xúc xắc:
 - Nếu kết quả là 1 (xác suất 1/6): thử một món ngẫu nhiên (explore).
 - Ngược lại (5/6 lần), chọn món đã biết ngon nhất (exploit). 📄

8. Ví dụ mô phỏng bằng Python (trong slide)

- Slide đưa ra một đoạn mã Python minh họa cách cài đặt **epsilon-greedy** trên môi trường multi-armed bandit.
- **Cấu trúc chính của mã:**
 1. **Khởi tạo** số cần gạt (arms) và giá trị thực tiềm năng $q_*(a)$.
 2. Cho agent tương tác qua nhiều vòng (episodes/steps).
 3. Tại mỗi bước:
 - Sinh số ngẫu nhiên $\in [0, 1]$.
 - Nếu $\leq \varepsilon$: chọn action ngẫu nhiên.
 - Ngược lại: chọn action có ước lượng $Q(a)$ lớn nhất.
 4. Nhận reward từ môi trường, sau đó **cập nhật** ước lượng $Q(a)$ bằng sample-average hoặc phương pháp incremental update.

5. Lập bảng thống kê tổng reward trung bình và tỷ lệ chọn mỗi cần gạt để đánh giá hiệu quả. 📄

• **Kết quả:**

- Làm từ từ, sau nhiều lần lặp, agent dần tập trung chọn arm có giá trị cao nhất, đồng thời vẫn giữ thói quen "thử" một số arm khác để cập nhật thông tin.
- Nếu ε quá lớn, agent sẽ khám phá quá mức, tổng reward giảm; nếu ε quá nhỏ, agent dễ rơi vào khai thác sớm, có thể bỏ lỡ arm tốt hơn. 📄

9. Tóm tắt lại

1. Khám phá (Exploration)

- Ưu tiên thu thập thông tin về môi trường/hành động.
- Giúp phát hiện các hành động chưa biết có thể cho reward cao hơn.
- Ví dụ: thử tất cả món trong nhà hàng theo Round Robin. 📄

2. Khai thác (Exploitation)

- Ưu tiên hành động có ước lượng giá trị cao nhất để tối đa hóa reward ngay.
- Ví dụ: luôn chọn món ngon nhất đã biết mỗi lần đến. 📄

3. Tradeoff

- Không thể vừa tối đa reward ngay (exploitation) vừa hoàn toàn thu thập thông tin (exploration).
- Cần chiến lược cân bằng ổn định để không bỏ lỡ hành động tiềm năng nhưng vẫn thu được reward đủ tốt. 📄

4. Chính sách epsilon-greedy

- Cung cấp cách đơn giản và hiệu quả để thực hiện tradeoff.
- Định kỳ (xác suất ε) thử ngẫu nhiên, còn lại khai thác action có $Q(a)$ cao nhất. 📄

Gợi ý mở rộng:

- Ngoài epsilon-greedy, còn có các phương pháp khác như Upper-Confidence-Bound (UCB) và Thompson Sampling, vốn cân nhắc mức độ "chưa chắc chắn" ở mỗi action để tự động điều chỉnh tần suất khám phá.
- Trong môi trường phi tĩnh (nonstationary), ta có thể giảm dần ε theo thời gian hoặc dùng phương pháp cấp phát bước (step-size) cố định để ước lượng nhanh thích ứng với sự thay đổi.